



# Linux

Trener: Michał Michalczuk

Gdańsk, 9 kwietnia 2017

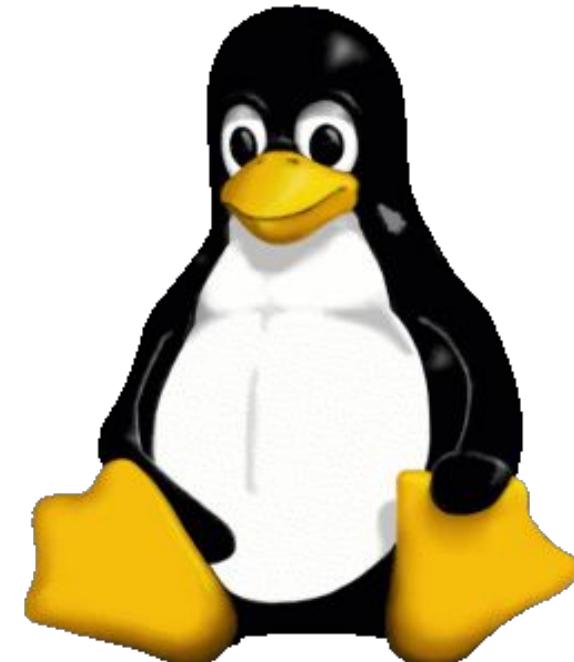
# Plan na zajęcia

- Linux – co to jest, skąd się wzięło, po co mi to?
- Jak się poruszać po Linux-ie? Struktura i konsola
- Uprawnienia
- Instalujemy pakiety, czyli programy
- Praca z plikami i potoki
- Przydatne narzędzia

# Czym Linux?

Linux - rodzina uniksopodobnych systemów operacyjnych opartych na jądrze Linux. Linux jest jednym z przykładów wolnego i otwartego oprogramowania (FOSS).

- Unix?
- Jądro Linux?



# Unix

*"... is a family of multitasking, multiuser computer operating systems"*

- Napisany w C
- Bell Labs od 1969
- W użyciu komercyjnym i akademickim od 1973

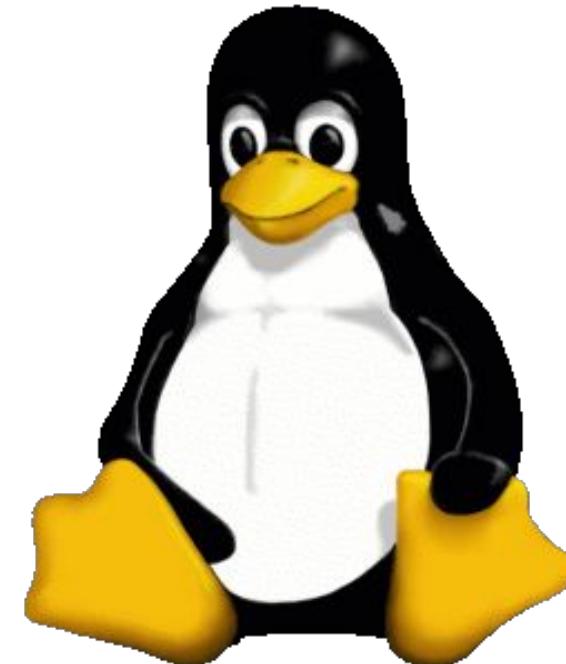


# Popularne UNIX-y



# Jądro (ang. *kernel*) Linux

**Jądro Linux** najważniejsza część napisana przez Linusa Torvaldsa w 1991 roku, a obecnie rozwijana przez licznych programistów z całego świata w ramach The Linux Foundation.



# Linus Torwalds

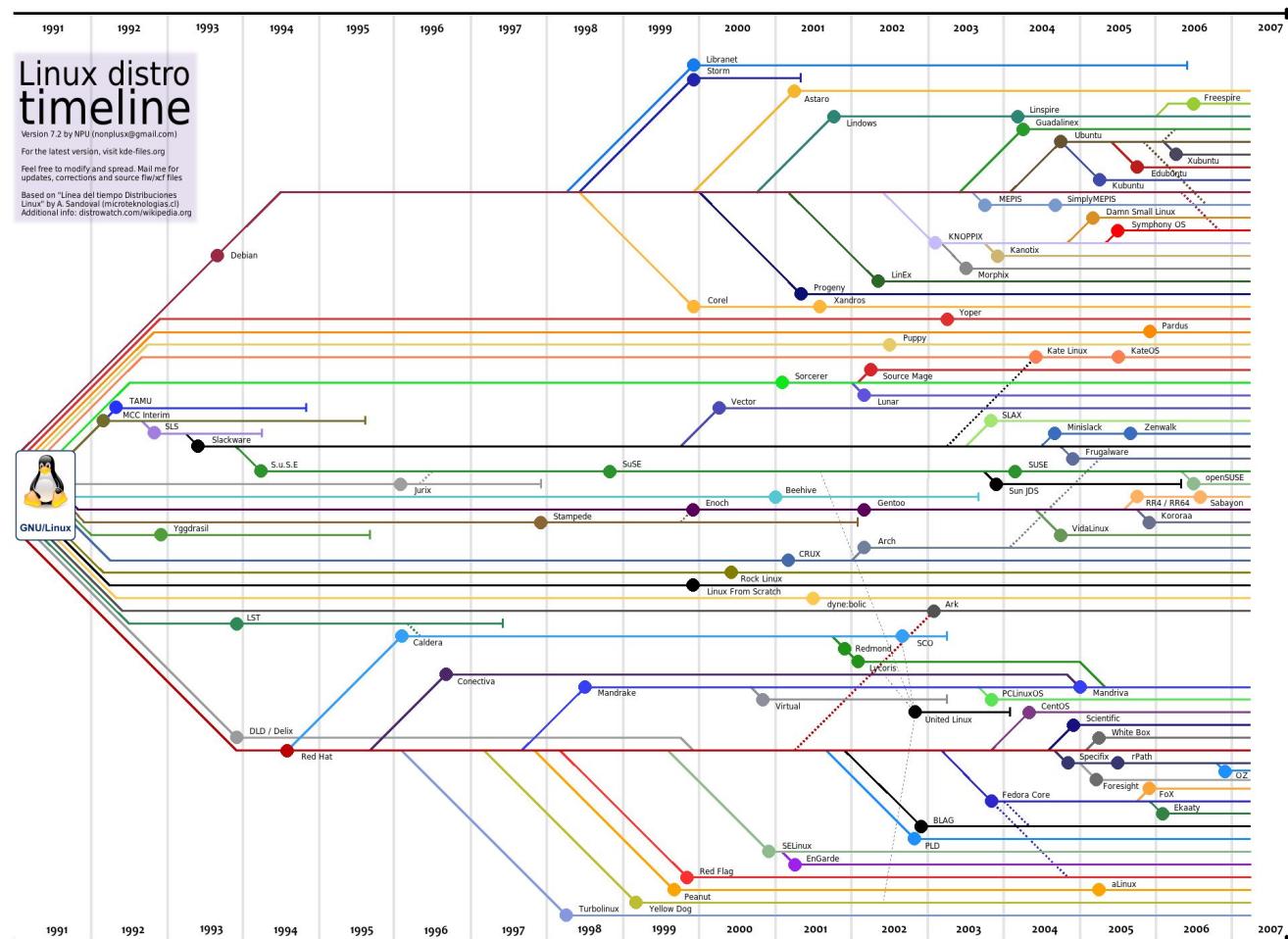
- 1991 Linux kernel
- 2005 Git

Linux = Linus + Unix



# Dystrybucje Linux

Drzewko z 2017 r. by się tutaj nie zmieściło :)



# Dystrybucje Linux



[teiatech.com.br](http://teiatech.com.br)

# Dystrybucje Linux

Android jest oparty o Linux kernel.



# Dystrybucje Linux

I ciągle powstają nowe.

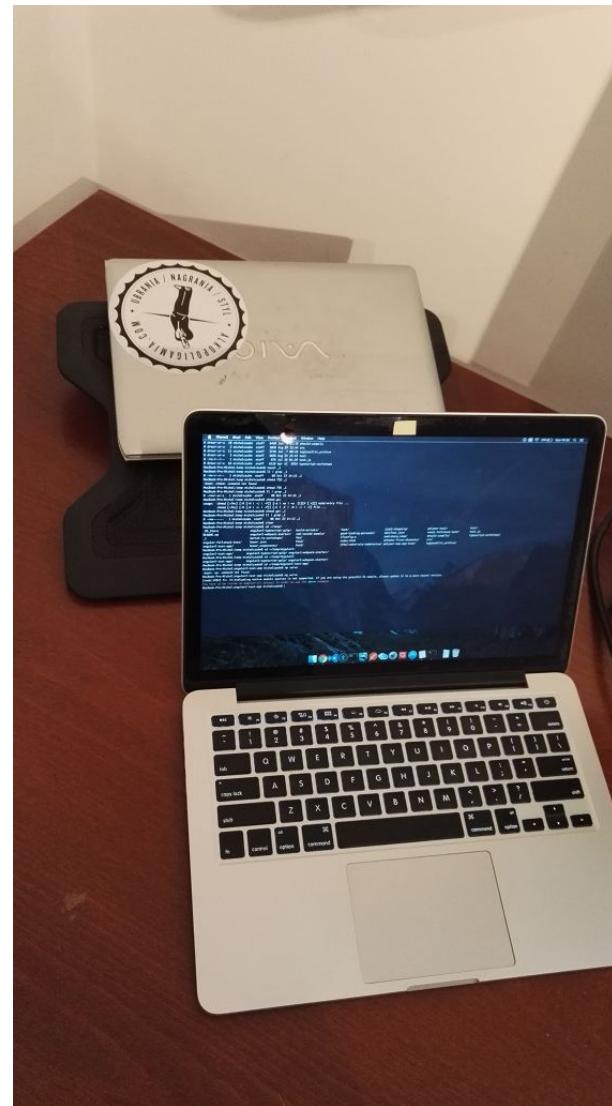
Najczęściej są bardzo dedykowane:

- Hostowanie aplikacji webowych
- Rourery
- Bezpieczeństwo
- Testy penetracyjne
- ...

Ubuntu – pochodna Debian-a

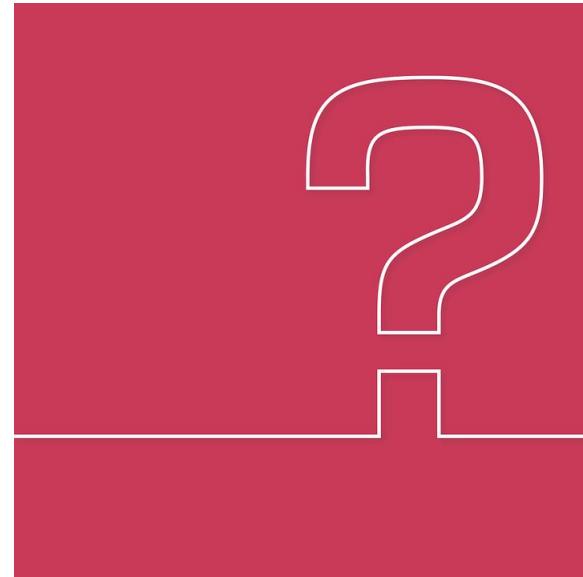


# Linux / Unix w domu i w zagrodzie

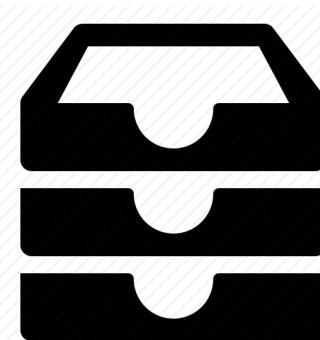


# Po co ten wstęp?

- Bo LINUX != Ubuntu
- To ogromny wybór systemów do różnych zastosowań
- Windows wcale nie jest tak potrzebny :)



# Struktura folderów i katalogów



# Odnośnie plików - Linux jest **case sensitive**

Czyli wrażliwy na wielkość liter.  
Windows jest niewrażliwy.

W praktyce?



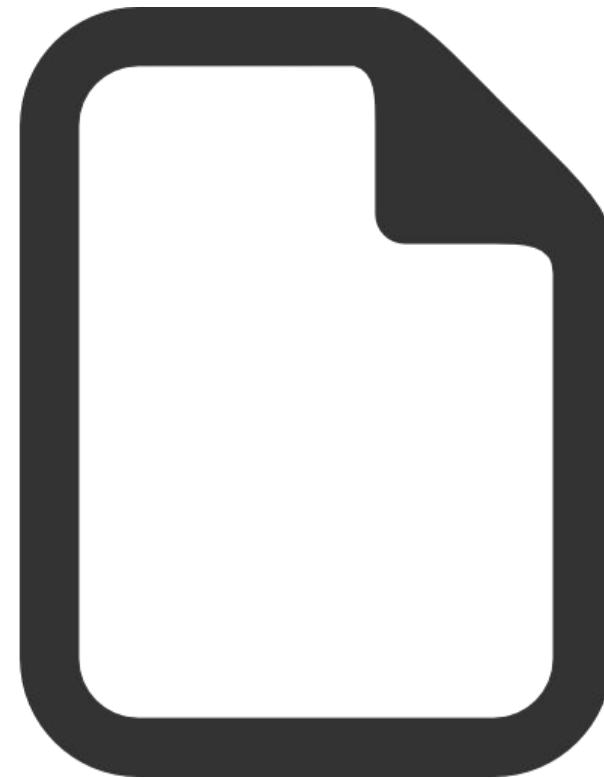
**Linux:**

*some\_file.txt* | *Some\_file.txt*      ==>      Dwa różne pliki

**Windows:**

*some\_file.txt* | *Some\_file.txt*      ==>      Ten sam plik

Wszystko jest plikiem ... wszystko



## [Showcase] Położenie kursora myszki - odczyt z pliku

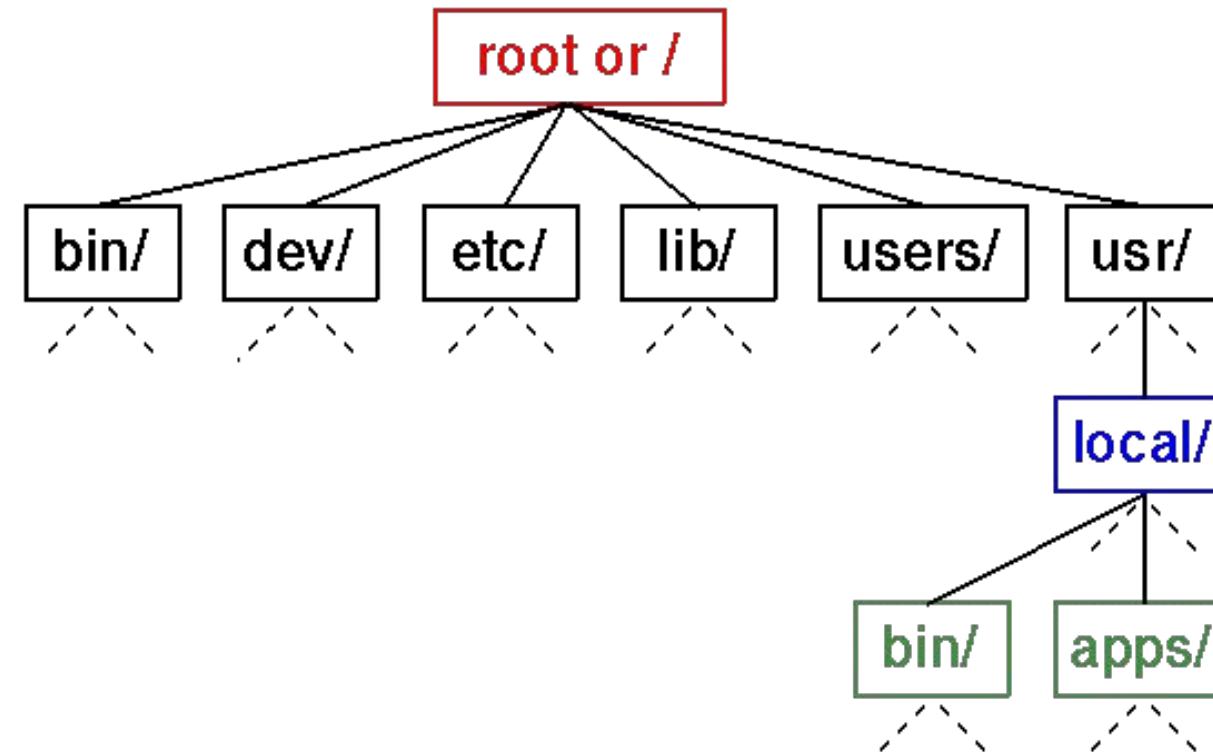
Nie, nie nie ... nie wpisujcie tego :)

*sudo cat /dev/input/event6*

cat - program do odczytu zawartości plików



# FHS (File Hierarchy Standard)



Więcej: [struktura katalogów](#)

# Podstawowa struktura katalogów

/ - root

**/home** - katalogi domowe użytkowników

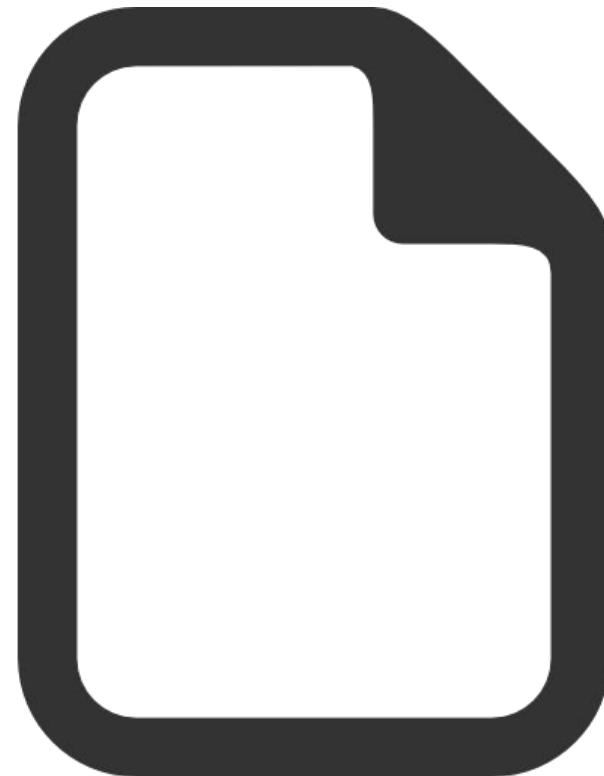
**/root** - katalog domowy administratora

**/usr** - programy użytkowników

**/etc** - ustawienia systemowe .. taki panel sterowania

# Wszystko jest plikiem ... wszystko

Pamiętacie ?



# Podstawowa struktura katalogów

**/dev** - urządzenia, pseudopliki o specjalnych uprawnieniach

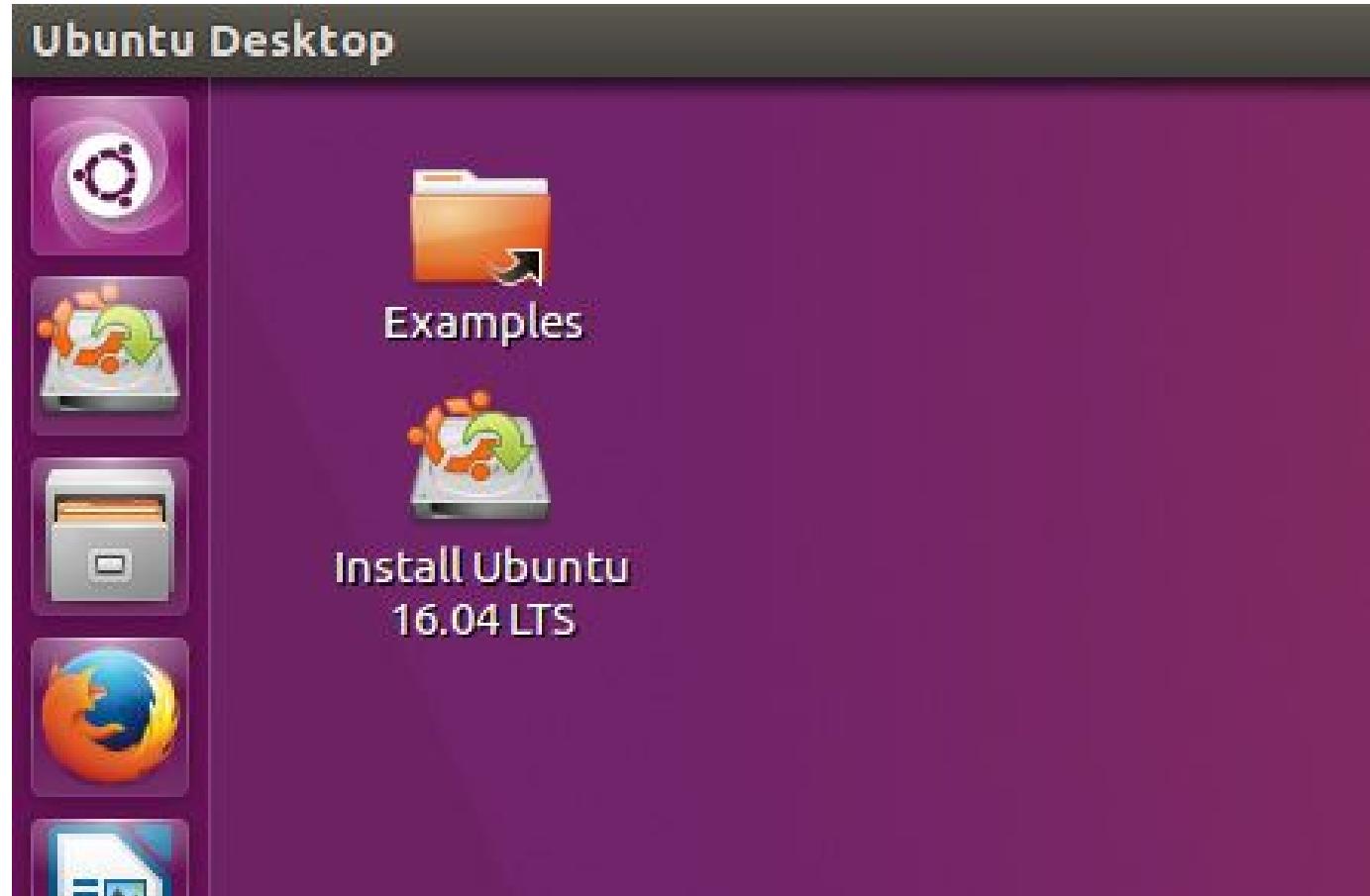
**/media** - zamontowane dyski, napęd CD, pendrive

**/var** - logi systemowe i logi wszelakie

**/proc** - dane o zasobach i aktualne wykonywanych programach, pseudopliki o specjalnych uprawnieniach

**~** - katalog domowy użytkownika

# Jak się po tym poruszać?



Albo ...

# Jak się po tym poruszać?

```
[MacBook-Pro-Michal:temp michalczukm$ ll
total 48
8 -rw-r--r--  1 michalczukm  staff  867B Aug 24 23:44 README.md
0 drwxr-xr-x  28 michalczukm  staff  952B Jul  2 21:04 angular-fullstack-test
0 drwxr-xr-x  14 michalczukm  staff  476B Jun 14 00:36 angular2-test-app
0 drwxr-xr-x  15 michalczukm  staff  510B May 10 01:34 angular2-typescript-gulp
0 drwxr-xr-x  21 michalczukm  staff  714B Jun  4 18:59 angular2-webpack-starter
0 drwxr-xr-x  3 michalczukm  staff  102B Apr 27 17:00 bartek-ts-workshops
8 -rw-r--r--  1 michalczukm  staff  419B Aug 24 23:44 bower.json
0 drwxr-xr-x  45 michalczukm  staff  1.5K Aug 24 23:44 bower_components
0 drwxr-xr-x  3 michalczukm  staff  102B Apr 10 2016 build-scripts
0 drwxr-xr-x  10 michalczukm  staff  340B Aug 27 20:04 es6-vscode-sample
0 drwxr-xr-x  5 michalczukm  staff  170B Sep  9 01:25 foo
0 drwxr-xr-x  3 michalczukm  staff  102B Aug  1 08:15 foo2
0 drwxr-xr-x  7 michalczukm  staff  238B Sep 16 16:41 foo3
0 drwxr-xr-x  3 michalczukm  staff  102B Oct 16 20:44 foo4
0 drwxr-xr-x  12 michalczukm  staff  408B Jun 20 04:59 good-looking-personal
8 -rw-r--r--  1 michalczukm  staff  1.9K Jun 12 22:38 ifconfig-a
8 -rw-r--r--  1 michalczukm  staff  412B Aug 24 23:44 index.html
0 drwxr-xr-x  6 michalczukm  staff  204B Jul 31 01:05 jfdz2-materialy-codereview
0 drwxr-xr-x  8 michalczukm  staff  272B Sep 18 16:31 jjdz2-shopping
8 -rw-r--r--  1 michalczukm  staff  134B Aug 24 23:44 manifest.json
0 drwxr-xr-x  3 michalczukm  staff  102B Apr 12 2016 omnisharp_temp
0 drwxr-xr-x@ 8 michalczukm  staff  272B Dec 11 2015 polymer-first-elements
0 drwxr-xr-x  9 michalczukm  staff  306B Aug 27 17:26 polymer-map-app-test
0 drwxr-xr-x  16 michalczukm  staff  544B Aug 24 00:22 polymer-test
0 drwxr-xr-x  23 michalczukm  staff  782B Jul  2 20:44 react-fullstack-test
0 drwxr-xr-x  19 michalczukm  staff  646B Jun  4 21:35 should-compile
0 drwxr-xr-x  3 michalczukm  staff  102B Aug 24 23:44 src
0 drwxr-xr-x  11 michalczukm  staff  374B Jun  7 00:28 tagtosellit_archive
0 drwxr-xr-x  3 michalczukm  staff  102B Aug 24 23:44 test
8 -rw-r--r--  1 michalczukm  staff  87B Jul 26 01:29 test.js
0 drwxr-xr-x  18 michalczukm  staff  612B Apr 12 2016 typescript-workshops
MacBook-Pro-Michal:temp michalczukm$ cd angular2-webpack-starter/
```

# Parę przydatnych komend

**pwd** - (provide working directory) czyli zobacz gdzie jesteś

np.:

*pwd*

**PS:**

~ - oznaczenie katalogu domowego użytkownika

# Parę przydatnych komend

**cd {path}** - przejdź do ścieżki

*cd ..* - przejdź katalog wyżej

*cd .* - przejdź .. tam gdzie teraz jesteś

*cd ~* - przejdź do twojego katalogu domowego

np.:

*cd Pictures/*

*cd ~/Pictures/*

# Parę przydatnych komend

## clear - wyczyść ekran

```
total 128
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 .
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 ..
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 metadata_spec.ts
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 provider_spec.ts
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 viewChild
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 e2e_test
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 view_child_spec.ts
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 module.ts
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 view_child_example.ts
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 view_child_howto.ts
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 viewChildren
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 e2e_test
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 view_children_spec.ts
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 module.ts
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 view_children_example.ts
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 view_children_howto.ts
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 ts
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 bootstrap
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 metadata
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 lifecycle_hooks_spec.ts
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 metadata.ts
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 platform
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 platform.ts
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 prod_mode
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 my_component.ts
drwxrwxr-x 1 michalczukm 1000  4096 May 15 10:00 prod_mode_example.ts
-rw-rw-r-- 1 michalczukm 1000    0  May 15 10:00 long-file.txt
-rw-rw-r-- 1 michalczukm 1000    0  May 15 10:00 README.md

24 directories, 32 files
michalczukm@michalczukm-VirtualBox:~/linux_classes$ clear
```

```
michalczukm@michalczukm-VirtualBox:~/linux_classes$ █
```

# Parę przydatnych komend

**ls {path}** - pokaż listę plików

**ls ..** - pokaż listę plików katalog wyżej

**ls .** - pokaż listę plików .. tam gdzie teraz jesteś

**ls ~** - pokaż listę z twojego katalogu domowego

np.:

*ls Pictures/*

*ls ~/Pictures/*

# [Console] Strike 1 – poruszanie się po katalogach

0. Wypakuj pliki ćwiczenia w katalogu domowym, przez interfejs okienkowy :)
1. Zaczynając od katalogu domowego ( **cd ~** ) przejdź do tych katalogów:

- A) **linux\_classes/core/debug**
- B) **linux\_classes/core**
- C) **linux\_classes/core/di**

(!) Po każdym punkcie wróć do ~ (użyj **cd ~**)

---



2. Zaczynając teraz od katalogu **~/linux\_classes/core/di/ts**, przejdź do poniższych katalogów. Używaj teraz tylko **ścieżek względnych !!**

- A) **linux\_classes/core/debug**
- B) **linux\_classes/core**
- C) **linux\_classes/core/di**

(!) Po każdym punkcie wróć do **~/linux\_classes/core/di/ts** (użyj **cd ~/linux\_classes/core/di/ts**)

## Mało wygodne to `ls`

// {path} - pokaż listę plików

// .. - pokaż listę plików katalog wyżej

// . - pokaż listę plików .. tam gdzie teraz jesteś

// ~ - pokaż listę z twojego katalogu domowego

Zobacz teraz jak wyglądają twoje pliki w ~

## Output II – o co tu chodzi?

```
drwxr-xr-x 15 michalczukm michalczukm 4096 paź 23 13:06 -/
drwxr-xr-x  3 root          root        4096 paź 18 00:00 ../
-rw-----  1 michalczukm michalczukm   107 paź 18 00:13 .bash_history
-rw-r--r--  1 michalczukm michalczukm   220 paź 18 00:00 .bash_logout
-rw-r--r--  1 michalczukm michalczukm 3771 paź 18 00:00 .bashrc
drwx----- 12 michalczukm michalczukm 4096 paź 23 13:21 .cache/
drwx----- 15 michalczukm michalczukm 4096 paź 18 00:09 .config/
drwxr-xr-x  2 michalczukm michalczukm 4096 paź 18 00:06 Desktop/
```

# Output II – o co tu chodzi?

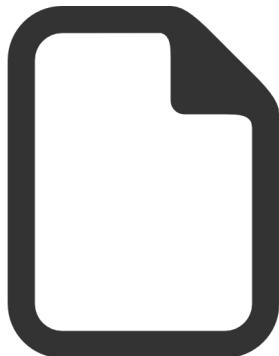
typ (-,d,l)	prawa dostępu	liczba linków	właściciel	grupa	wielkość ( w Byte)	TimeStamp	Nazwa pliku/katalogu
d	rwxr-xr-x	15	michalczukm	michalczukm	4096	paź 23 13:06	./
d	rwxr-xr-x	3	root	root	4096	paź 18 00:00	../
-	rw-----	1	michalczukm	michalczukm	107	paź 18 00:13	.bash_history
-	rw-r--r--	1	michalczukm	michalczukm	220	paź 18 00:00	.bash_logout
-	rw-r--r--	1	michalczukm	michalczukm	3771	paź 18 00:00	.bashrc
d	rwx-----	11	michalczukm	michalczukm	4096	paź 23 13:06	.cache/
d	rwx-----	15	michalczukm	michalczukm	4096	paź 18 00:09	.config/
d	rwxr-xr-x	2	michalczukm	michalczukm	4096	paź 18 00:06	Desktop/

- : plik

d : katalog

l : link

# Tworzenie plików i katalogów



# Tworzenie katalogu

**mkdir {dir name}** - stwórz katalog o nazwie *{dir name}*

*mkdir ../{dir name}* - stwórz go katalog wyżej

*mkdir ~/ {dir name}* - stwórz go w katalogu domowym

np.:

*mkdir temp*

*mkdir ~/temp*

**! Pliki i katalogi nazywamy bez spacji !**

## Tworzenie pliku

**touch {file name}** - stwórz plik o nazwie *{file name}*

*touch ../{file name}* - stwórz go katalog wyżej

*touch ~/ {dir name}* - stwórz go w katalogu domowym

np.:

*touch ~/temp/my\_new\_file*

*touch /tmp/this\_will\_be\_remove\_at\_reboot*

**! Pliki i katalogi nazywamy bez spacji !**

# Usuwanie

## Plik:

**rm {file name}** - usuń plik o nazwie *{file name}*

Oczywiście działa też dla ścieżek względnych :)

(!) Wiele plików naraz

*rm {file-1} {file-2}*

## Katalog:

**rm -r {dir name}** - usuń katalog o nazwie *{dir name}*

Parametry:

**-r** (recursive)

Zanim przejdziemy do ćwiczenia

`sudo apt install tree`

`tree {path}`

## [Console] Strike 2 – tworzenie plików

```
~/linux_classes
|
|--- animals
|   |
|   |--- dogs
|   |   |--> albert.anim
|   |
|   |--- cats
|   |   |--> fluffy.anim
|   |
|--- beasts
|   |
|---> beast_list.txt
```



# Pomoc wbudowana w Linux

**HELP  
WANTED**

# Pamiętacie II do wyświetlania listy plików ?

**II {path}** - to tak naprawdę program **ls**.

Tylko, że z paroma parametrami.

**ls –lsa {path}**

Parametry:

- l (long listing format)
- s (print size)
- a (show all, do not ignore files starting with .)

Mhm .. a skąd znasz te parametry?

- man
- --help



# man (manual)

**man {program}** - wyświetla manual

- programu
- polecenia systemowego
- pseudoplików (jak w /dev)
- wywołania plików bibliotek
- itd.

Np.: *man ls*

{program} --help

**Szybka pomoc.**

Mniej rozbudowany niż **man**.

Główne parametry i ich znaczenie.

Zajeży od dobrej woli programisty :) To tylko konwencja.

## [Console] Strike 2 – pomoc

Zobaczmy co można zrobić w programie `tree`.

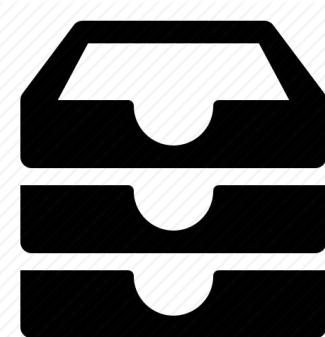
**Znajdź w dokumentacji `tree` jak wyświetlić wszystkie uprawnienia (permissions) plików i folderów.**



Rzuć potem okiem na dokumentację `chmod`. Też się przyda.

Tak. To będzie potrzebne do kolejnego ćwiczenia :)

# Uprawnienia



# Uprawnienia - Jak to wygląda?

*drwxrwxrwx dir\_name*

*d rwx rwx rwx dir\_name*

*dir read/write/execute rwd rwd dir\_name*

# User / Group / Others

*r w x*

*Read Write Execute*

A dlaczego są 3 takie zestawy?

**rwX**

**user**

**rwX**

**group**

**rwX**

**others**

# Kiedy mamy uprawnienie?

*0 – brak uprawnienia*

*1 – jest uprawnienie*

*r w x*      ---->      111

*- - -*      ---->      000

*r - x*      ---->      101

*- - x*      ---->      001

# Kiedy mamy uprawnienie?

*Przykłady:*

- |                      |                     |
|----------------------|---------------------|
| <i>d rwx r-x --x</i> | <i>my_temp_dir</i>  |
| <i>- r-x -w- ---</i> | <i>my-note.txt</i>  |
| <i>- rwx r-x r-x</i> | <i>my_script.sh</i> |

# chmod (*change mode*)

**chmod {options} {file/dir name}** - zmień uprawnienia

Jest tutaj parę możliwości.

u – user  
g – group  
o - others

np.:

chmod g+w my\_file.txt

chmod go-w my\_file.txt

chmod go= my\_file.txt

chmod 700 my\_file.txt



# chmod (*change mode*) dla folderu i całej zawartości

**chmod –R {permissions} {dir name}** - zmień uprawnienia

Jest tutaj parę możliwości.

np.:

chmod -R g+w my\_file.txt

chmod -R go-w my\_file.txt

chmod -R go= my\_file.txt

chmod -R 700 my\_file.txt

# Uprawnienia do plików i katalogów - więcej

<http://www.cybertech.net.pl/online/system/linuxzin/pliki/atrib.htm>

<https://pl.wikipedia.org/wiki/Chmod>

# [Console] Strike 4 – uprawnienia

```
~/linux_classes
|
---- animals (rwx--xrw-)
    |
    ----dogs (rwx--xrw-)
        |
        |--> albert.anim (rwx-----)
        |
        ----cats (rwx-w---x)
            |
            |--> fluffy.anim (r-x--xrwx)
            |
---- beasts (rwx-----)
    |
    ----> beast_list.txt (rwx-----)
```

## Hint

Spróbuj ponownie użyć `tree`.

Ale tym razem `tree -p`



# sudo – Super User Do

**sudo {komenda}** - wykonaj daną komendę jako super user (admin)

np.:

sudo chmod -R g+w my\_file.txt

sudo apt-get install tree

# Pakiety



# apt – Advanced Packaging Tool

**apt** to zestaw programów i komend do zarządzania pakietami.

Pod spodem korzysta z **dpkg** (debian package manager).



## apt a apt-get

Aktualnie wszystko jest przykryte fasadą “apt”.

Jeszcze pewien czas temu było wiele oddzielnych programów.

Więc.

**Jeśli nie działa Ci jakieś polecenie *apt* spróbuj użyć *apt-get*.**

# apt install – instalacja pakietu

**apt install {package-name}** - zainstaluj pakiet o danej nazwie.

np.:

apt install tree

apt install guake



# apt-get remove – usunięcie pakietu

**apt-get remove {package-name}** - usuń pakiet o danej nazwie.

**apt-get remove --purge {package-name}** - usuń pakiet z jego konfiguracją

np.:

apt-get remove tree

apt-get remove guake



# apt-get remove – nie usuwa wszystkiego

**apt-get remove {package-name}** nie usuwa zależności, które pozostały po instalacji :(

**apt-get autoremove** - usuń "sieroty" z systemu.



Sieroty – nie używane pakiety.

# **dpkg -l : zobacz listę zainstalowanych pakietów**

**dpkg -l**

Pokaż listę wszystkich zainstalowanych pakietów w systemie.



# Hmm, skąd w ogóle te programy?

Z internetu :)

Plik: **/etc/apt/sources.list**

Zawiera listę repozytoriów.



## Inne przydatne komendy apt-get

**apt-cache search {what-you're-looking-for}**

Znajdź pakiet

**apt-get update**

Zaktualizuj listę programów (wg repozytoriów)



## [Console] Strike 5 – pakiety

1.

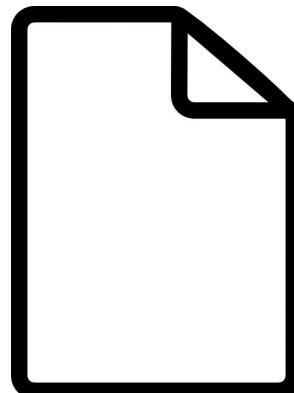
Znajdź (apt-search)  
Zainstaluj (apt-get install),  
oraz przetestuj (uruchom) pakiety:

- **guake**
- tree
- mc (midnight commander)

2. Usuń (ale włącznie z konfiguracją i "wyczyść" potem przez apt) pakiet **guake**



# Praca z plikami i tekstem



## echo - zwróć tekst

**echo "your message"**

Efektem działania będzie wyświetlenie wiadomości na ekranie.

```
$> echo "some first message"
```

```
$> some first message
```



## cat - czytaj zawartość pliku

`cat {file_name}`

Efektem działania będzie wyświetlenie zawartości pliku `{file_name}` na ekranie.

```
$> cat my_awesone_file
```

*My awesome file awesome content.*

*So cool. Wow.*



# less- czytaj zawartość strona po stronie

**less {file\_name}**

Efektem działania będzie wyświetlenie zawartości pliku *{file\_name}* na ekranie, ale podzielonego na strony.

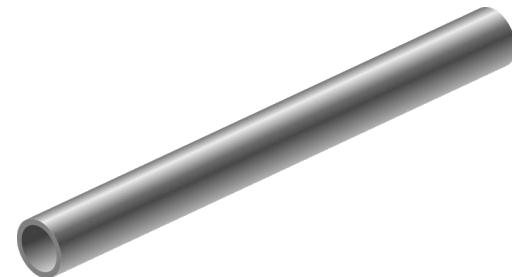


## [Console] Strike 6 – praca z tekstem

1. Wypisz na ekranie tekst “Echoed some text”
2. Wypisz na ekranie adres katalogu użytkownika.  
(hint, jest zmienna systemowa \$HOME\$)
3. Odczytaj plik `long-file.txt` za pomocą less



# Pipes and redirections | potoki i przekierowania



## Pipe – potoki ( | )

Potoki przekazują wyjęcie jednego programu jako wejście 2giego programu.

Np.:

**ls --help | less**

(wyświetl pomoc dla programu **ls** i przekaż tekst do programu **less**)



## Przekierowanie (>) oraz (>>)

Przekierowywujemy wynik działania programu do pliku.  
Jeśli go nie ma - stworzymy nowy.

Jeśli chcemy dopisać do istniejącego pliku. Ale go nie zastąpić, to używamy >>

Np.:

**echo “nanana” > my\_file.txt**

Np.:

**echo “nanana” >> my\_file.txt**

# Przekierowanie (<)

Tym razem przekierowywujemy wynik działania z pliku. do programu.

Np.:

```
wc -l < my_file.txt
```

“*wc -l*” - zlicz linie

“*wc -w*” - zlicz słowa

# Ściągawka

Symbol	Znaczenie
>	Zapisz wyjście programu do pliku
>>	Dołącz wyjście programu do pliku
<	Odczytaj wejście programu z pliku
	Wyślij wyjście jednego programu jako wejście drugiego

# Parę przydatnych komend do Ćwiczenia

**sort** - sortuje zawartość pliku

np.: *sort {file path}*

**head** - wyświetla wskazaną ilość pierwszych wpisów

np.: *head -5 README.md*

oczywiście zamiast 5 możesz wpisać dowolną liczbę

## [Console] Strike 7 – pipes and redirections

0) Odpalmy przykłady z prezentacji.

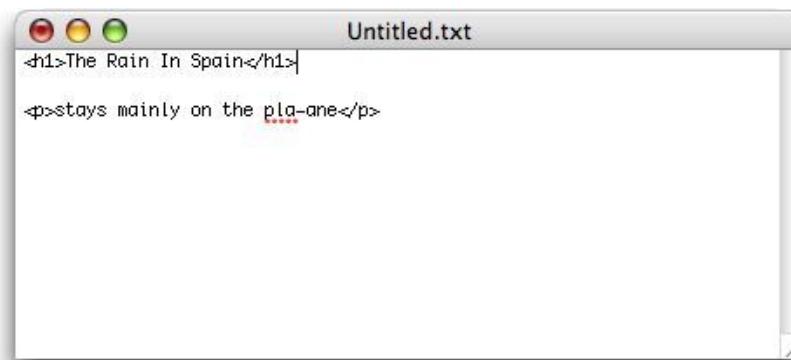
Potem przejdź do folderu ***linux\_classes***

Używając pipe-ów (|) i przekierowań (>), wykonaj:

- 1) wyświetl tree, posortuj jego wyniki i wyświetl przez less
- 2) wyświetl tree, posortuj **odwrotnie** jego wyniki i wyświetl przez less
- 3) wyświetl tree i wyświetl ilość linii które tam są (użyj wc)
- 4) wyświetl tree, ale tylko 5 pierwszych elementów (użyj head), potem posortuj ten wynik i zapisz do pliku result\_4.txt



# Edytory tekstu: mcedit, nano, vim



# Tools: zip, grep, ps



# grep - wyszukaj w tekście

Zaawansowany program do wyszukiwania w tekście.

Użycie:

***grep {pattern} {options} file/files***

np.: grep angular README.md

```
michalczukm@michalczukm-VirtualBox:~/linux_classes$ grep code README.md
Source files in this exercise are part of source code from `angular` framework repository.
## Full source code
For check the full source code please visit [https://github.com/angular/angular](https://github.com/angular/angular)
michalczukm@michalczukm-VirtualBox:~/linux_classes$ grep angular README.md
Source files in this exercise are part of source code from 'angular' framework repository.
For check the full source code please visit [https://github.com/angular/angular](https://github.com/angular/angular)
michalczukm@michalczukm-VirtualBox:~/linux_classes$
```

# ps - wyświetl listę procesów w systemie

Tak samo jak w windowsie mamy “Task Manager”, tak mamy ps w linuxie.

Użycie:

**ps**

Aby zobaczyć wszystkie procesy, ich identyfikatory oraz właściciela:

**ps -aux**

Parametry:

- a : wszystkie moje procesy
- u : pokaż usera który uruchomił
- x : wszystkie inne procesy

# top - resource manager

Wyświetl procesy i pokaż ile zasobów zużywają

Uzycie:

**top**

```
top - 00:44:21 up 12:20,  3 users,  load average: 0,21, 0,18, 0,18
Tasks: 206 total,  1 running, 205 sleeping,  0 stopped,  0 zombie
%Cpu(s): 3,3 us, 1,2 sy, 0,0 ni, 95,5 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 6136636 total, 548344 free, 1507308 used, 4080984 buff/cache
KiB Swap: 6317052 total, 6317052 free,      0 used. 4226512 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2419	michalc+	20	0	1852776	638200	75392	S	8,3	10,4	100:36.49	compiz
1997	root	20	0	491624	181428	52364	S	3,7	3,0	10:52.78	Xorg
3241	michalc+	20	0	679300	47152	30664	S	0,7	0,8	0:47.40	gnome-terminal
2617	michalc+	20	0	1536360	289240	70516	S	0,3	4,7	0:52.63	gnome-software
4523	root	20	0	0	0	0	S	0,3	0,0	0:13.59	kworker/1:0
5414	michalc+	20	0	50112	4052	3364	R	0,3	0,1	0:00.01	top
1	root	20	0	119668	5984	4176	S	0,0	0,1	0:02.98	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.02	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.78	ksoftirqd/0
5	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/0:0H
7	root	20	0	0	0	0	S	0,0	0,0	0:04.96	rcu_sched
8	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_bh
9	root	rt	0	0	0	0	S	0,0	0,0	0:01.94	migration/0
10	root	rt	0	0	0	0	S	0,0	0,0	0:00.59	watchdog/0
11	root	rt	0	0	0	0	S	0,0	0,0	0:00.60	watchdog/1
12	root	rt	0	0	0	0	S	0,0	0,0	0:03.50	migration/1
13	root	20	0	0	0	0	S	0,0	0,0	0:12.38	ksoftirqd/1
15	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/1:0H
16	root	20	0	0	0	0	S	0,0	0,0	0:00.01	kdevtmpfs
17	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	netns
18	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	perf
19	root	20	0	0	0	0	S	0,0	0,0	0:00.04	khungtaskd
20	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	writeback
21	root	25	5	0	0	0	S	0,0	0,0	0:00.00	ksmd
22	root	39	19	0	0	0	S	0,0	0,0	0:03.36	khugepaged
23	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	crypto
24	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kintegrityd

# htop - top na sterydach

Wyświetl procesy i pokaż ile zasobów zużywają.

Uzycie:

**htop**

Nie znaleziono?

*sudo apt install htop*

```

 1 [|||] 0.7% Tasks: 118, 239 thr; 1 running
 2 [|||] 4.0% Load average: 0.25 0.19 0.18
Mem[|||||] 1.54G/5.85G Uptime: 12:19:44
Swp[          ] 0K/6.02G

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
2419 michalczu 20 0 1809M 623M 75392 S 2.7 16.4 1h40:32 compiz
2486 michalczu 20 0 1809M 623M 75392 S 1.3 10.4 41:37.58 compiz
2487 michalczu 20 0 1809M 623M 75392 S 1.3 10.4 41:34.51 compiz
1997 root 20 0 480M 177M 52364 S 1.3 3.0 10:51.33 /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /var/r
5409 michalczu 20 0 34928 4532 3260 R 0.7 0.1 0:00.07 htop
3241 michalczu 20 0 663M 47152 30664 S 0.7 0.8 0:47.06 /usr/lib/gnome-terminal/gnome-terminal-server
3198 michalczu 20 0 647M 27808 17596 S 0.0 0.5 0:01.24 /usr/bin/unity-scope-loader applications/application
2326 michalczu 20 0 360M 11100 7204 S 0.0 0.2 0:26.22 /usr/bin/ibus-daemon --daemonize --xim
Ubuntu Software 20 0 116M 5984 4176 S 0.0 0.1 0:02.98 /lib/systemd/systemd --system --deserialize 25
  ozo root 20 0 34880 3052 2592 S 0.0 0.0 0:00.60 /lib/systemd/systemd-journald
1775 syslog 20 0 250M 3364 2688 S 0.0 0.1 0:00.05 /usr/sbin/rsyslogd -n
1776 syslog 20 0 250M 3364 2688 S 0.0 0.1 0:00.01 /usr/sbin/rsyslogd -n
1777 syslog 20 0 250M 3364 2688 S 0.0 0.1 0:00.08 /usr/sbin/rsyslogd -n
1744 syslog 20 0 250M 3364 2688 S 0.0 0.1 0:00.17 /usr/sbin/rsyslogd -n
1778 root 20 0 329M 10736 7016 S 0.0 0.2 0:00.00 /usr/sbin/ModemManager
1782 root 20 0 329M 10736 7016 S 0.0 0.2 0:00.00 /usr/sbin/ModemManager
1757 root 20 0 329M 10736 7016 S 0.0 0.2 0:00.04 /usr/sbin/ModemManager
1764 root 20 0 4400 1352 1256 S 0.0 0.0 0:01.26 /usr/sbin/acpid
1767 messagebus 20 0 44412 5384 3604 S 0.0 0.1 0:02.85 /usr/bin/dbus-daemon --system --address=systemd: --r
1790 root 20 0 37296 3192 2904 S 0.0 0.1 0:00.09 /usr/sbin/cron -f
1828 root 20 0 292M 10736 7724 S 0.0 0.2 0:01.24 /usr/lib/accountsservice/accounts-daemon
1830 root 20 0 292M 10736 7724 S 0.0 0.2 0:00.02 /usr/lib/accountsservice/accounts-daemon
1793 root 20 0 292M 10736 7724 S 0.0 0.2 0:01.35 /usr/lib/accountsservice/accounts-daemon
1796 root 20 0 28628 3192 2812 S 0.0 0.1 0:00.25 /lib/systemd/systemd-logind
1800 whoopsie 20 0 451M 16288 13008 S 0.0 0.3 0:00.00 /usr/bin/whoopsie -f
1981 whoopsie 20 0 451M 16288 13008 S 0.0 0.3 0:00.02 /usr/bin/whoopsie -f
1798 whoopsie 20 0 451M 16288 13008 S 0.0 0.3 0:00.27 /usr/bin/whoopsie -f
1960 root 20 0 453M 17172 14408 S 0.0 0.3 0:00.08 /usr/sbin/NetworkManager --no-daemon
1962 root 20 0 453M 17172 14408 S 0.0 0.3 0:00.13 /usr/sbin/NetworkManager --no-daemon
1799 root 20 0 453M 17172 14408 S 0.0 0.3 0:01.04 /usr/sbin/NetworkManager --no-daemon
1800 avahi 20 0 44920 3156 2824 S 0.0 0.1 0:00.75 avahi-daemon: running [MichałczukM-VirtualBox.local]
1893 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.33 /usr/lib/snapd/snapd
1896 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.00 /usr/lib/snapd/snapd
1902 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.01 /usr/lib/snapd/snapd
1990 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.45 /usr/lib/snapd/snapd
1991 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.18 /usr/lib/snapd/snapd
1992 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.61 /usr/lib/snapd/snapd
5424 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.20 /usr/lib/snapd/snapd
13962 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.26 /usr/lib/snapd/snapd
13963 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.28 /usr/lib/snapd/snapd
1814 root 20 0 233M 15452 9940 S 0.0 0.3 0:02.44 /usr/lib/snapd/snapd
F1 Help F2 Setup F3 Search F4 Filter F5 Tree F6 Sort By F7 Nice F8 Nice + F9 Kill F10 Exit

```

# PID - Process Id

Identyfikator procesu.

Można go zobaczyć przez  
ps/top/htop.

To ten 4 cyfrowy numer.

```

 1 [|||||] 0.7% Tasks: 118, 239 thr; 1 running
 2 [|||||] 4.0% Load average: 0.25 0.19 0.18
Mem[|||||] 1.54G/5.85G Uptime: 12:19:44
Swp[|||||] 0K/6.02G

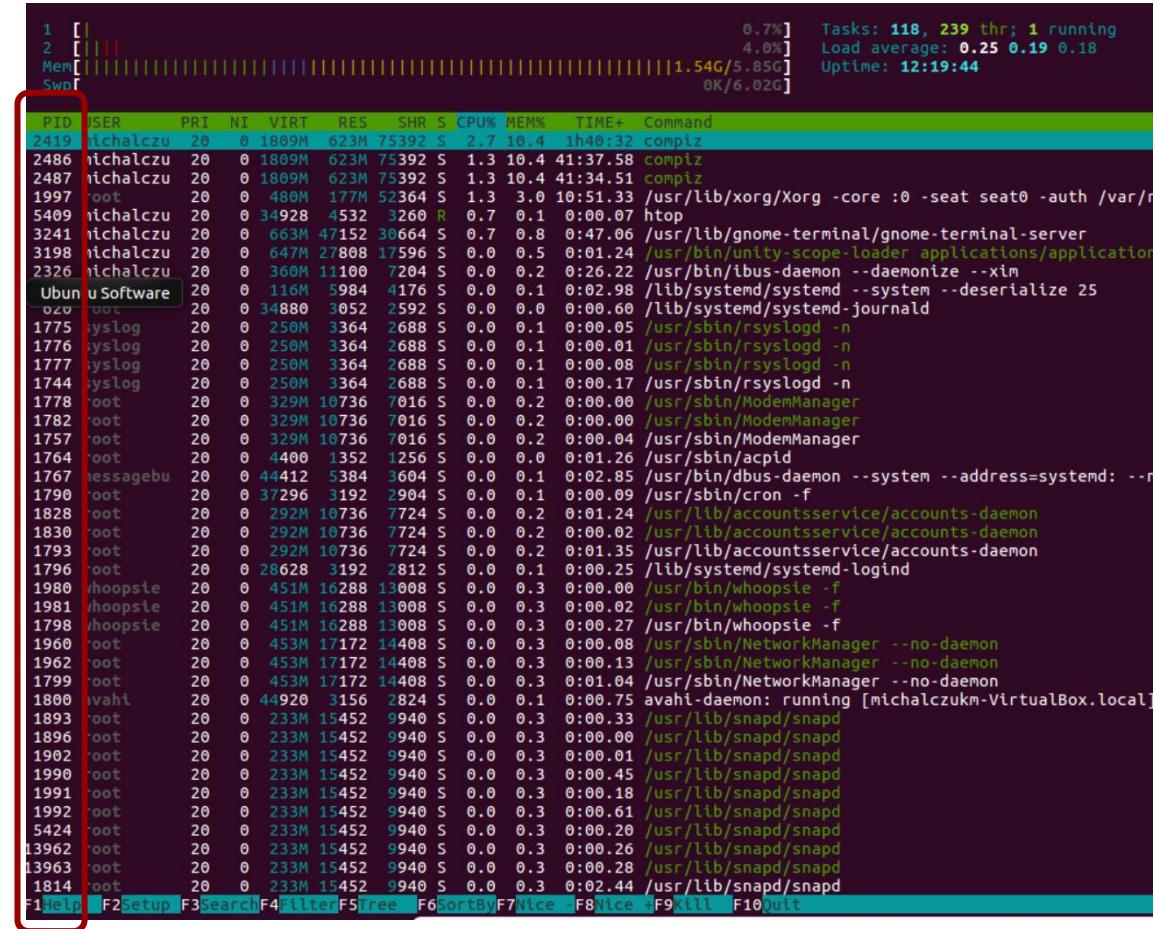
          PID  SER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
2419  michalczu 20  0 1809M 623M 75392 S 2.7 10.4 1h40:32 compiz
2486  michalczu 20  0 1809M 623M 75392 S 1.3 10.4 41:37.58 compiz
2487  michalczu 20  0 1809M 623M 75392 S 1.3 10.4 41:34.51 compiz
1997  root    20  0 480M 177M 52364 S 1.3 3.0 10:51.33 /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /var/r
5409  michalczu 20  0 34928 4532 3260 R 0.7 0.1 0:00.07 htop
3241  michalczu 20  0 663M 47152 30664 S 0.7 0.8 0:47.06 /usr/lib/gnome-terminal/gnome-terminal-server
3198  michalczu 20  0 647M 27808 17596 S 0.0 0.5 0:01.24 /usr/bin/unity-scope-loader applications/application
2326  michalczu 20  0 360M 11100 7204 S 0.0 0.2 0:26.22 /usr/bin/ibus-daemon --daemonize --xim
UbuntuSoftware 20  0 116M 5984 4176 S 0.0 0.1 0:02.98 /lib/systemd/systemd --system --deserialize 25
  ozo  root    20  0 34880 3052 2592 S 0.0 0.0 0:00.60 /lib/systemd/systemd-journald
1775  syslog   20  0 250M 3364 2688 S 0.0 0.1 0:00.05 /usr/sbin/rsyslogd -n
1776  syslog   20  0 250M 3364 2688 S 0.0 0.1 0:00.01 /usr/sbin/rsyslogd -n
1777  syslog   20  0 250M 3364 2688 S 0.0 0.1 0:00.08 /usr/sbin/rsyslogd -n
1744  syslog   20  0 250M 3364 2688 S 0.0 0.1 0:00.17 /usr/sbin/rsyslogd -n
1778  root    20  0 329M 10736 7016 S 0.0 0.2 0:00.00 /usr/sbin/ModemManager
1782  root    20  0 329M 10736 7016 S 0.0 0.2 0:00.00 /usr/sbin/ModemManager
1757  root    20  0 329M 10736 7016 S 0.0 0.2 0:00.04 /usr/sbin/ModemManager
1764  root    20  0 4400 1352 1256 S 0.0 0.0 0:01.26 /usr/sbin/acpid
1767  messagebus 20  0 44412 5384 3604 S 0.0 0.1 0:02.85 /usr/bin/dbus-daemon --system --address=systemd: --r
1790  root    20  0 37296 3192 2904 S 0.0 0.1 0:00.09 /usr/sbin/cron -f
1828  root    20  0 292M 10736 7724 S 0.0 0.2 0:01.24 /usr/lib/accountsservice/accounts-daemon
1830  root    20  0 292M 10736 7724 S 0.0 0.2 0:00.02 /usr/lib/accountsservice/accounts-daemon
1793  root    20  0 292M 10736 7724 S 0.0 0.2 0:01.35 /usr/lib/accountsservice/accounts-daemon
1796  root    20  0 28628 3192 2812 S 0.0 0.1 0:00.25 /lib/systemd/systemd-logind
1980  whoopsie  20  0 451M 16288 13008 S 0.0 0.3 0:00.00 /usr/bin/whoopsie -f
1981  whoopsie  20  0 451M 16288 13008 S 0.0 0.3 0:00.02 /usr/bin/whoopsie -f
1798  whoopsie  20  0 451M 16288 13008 S 0.0 0.3 0:00.27 /usr/bin/whoopsie -f
1960  root    20  0 453M 17172 14408 S 0.0 0.3 0:00.08 /usr/sbin/NetworkManager --no-daemon
1962  root    20  0 453M 17172 14408 S 0.0 0.3 0:00.13 /usr/sbin/NetworkManager --no-daemon
1799  root    20  0 453M 17172 14408 S 0.0 0.3 0:01.04 /usr/sbin/NetworkManager --no-daemon
1800  avahi   20  0 44920 3156 2824 S 0.0 0.1 0:00.75 avahi-daemon: running [MichałczukM-VirtualBox.local]
1893  root    20  0 233M 15452 9940 S 0.0 0.3 0:00.33 /usr/lib/snapd/snapd
1896  root    20  0 233M 15452 9940 S 0.0 0.3 0:00.00 /usr/lib/snapd/snapd
1902  root    20  0 233M 15452 9940 S 0.0 0.3 0:00.01 /usr/lib/snapd/snapd
1990  root    20  0 233M 15452 9940 S 0.0 0.3 0:00.45 /usr/lib/snapd/snapd
1991  root    20  0 233M 15452 9940 S 0.0 0.3 0:00.18 /usr/lib/snapd/snapd
1992  root    20  0 233M 15452 9940 S 0.0 0.3 0:00.61 /usr/lib/snapd/snapd
5424  root    20  0 233M 15452 9940 S 0.0 0.3 0:00.20 /usr/lib/snapd/snapd
13962 root    20  0 233M 15452 9940 S 0.0 0.3 0:00.26 /usr/lib/snapd/snapd
13963 root    20  0 233M 15452 9940 S 0.0 0.3 0:00.28 /usr/lib/snapd/snapd
1814  root    20  0 233M 15452 9940 S 0.0 0.3 0:02.44 /usr/lib/snapd/snapd
F1 Del F2 Setup F3 Search F4 Filter F5 Tree F6 Sort By F7 Nice F8 Nice + F9 Kill F10 quit

```

# kill - zamknij process

Gdy process się zawiesi/zakleszczy można go wyłączyć przez ***kill***.

- 1) ***kill {PID}***
- 2) ***killall -e {program-name}***



```

1 [|||||] 0.7% Tasks: 118, 239 thr; 1 running
2 [|||] 4.0% Load average: 0.25 0.19 0.18
Mem[|||||] 1.54G/5.85G Uptime: 12:19:44
Swp[|||||] 0K/6.02G

PID SER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
2419 michalczu 20 0 1809M 623M 75392 S 2.7 10.4 1h40:32 compiz
2486 michalczu 20 0 1809M 623M 75392 S 1.3 10.4 41:37.58 compiz
2487 michalczu 20 0 1809M 623M 75392 S 1.3 10.4 41:34.51 compiz
1997 root 20 0 480M 177M 52364 S 1.3 3.0 10:51.33 /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /var/r
5409 michalczu 20 0 34928 4532 3260 R 0.7 0.1 0:00.07 htop
3241 michalczu 20 0 663M 47152 30664 S 0.7 0.8 0:47.06 /usr/lib/gnome-terminal/gnome-terminal-server
3198 michalczu 20 0 647M 27808 17596 S 0.0 0.5 0:01.24 /usr/bin/unix-scope-loader applications/application
2326 michalczu 20 0 360M 11100 7204 S 0.0 0.2 0:26.22 /usr/bin/ibus-daemon --daemonize --xim
Ubuntu Software 20 0 116M 5984 4176 S 0.0 0.1 0:02.98 /lib/systemd/systemd --system --deserialize 25
020 root 20 0 34880 3052 2592 S 0.0 0.0 0:00.60 /lib/systemd/systemd-journald
1775 syslog 20 0 250M 3364 2688 S 0.0 0.1 0:00.05 /usr/sbin/rsyslogd -n
1776 syslog 20 0 250M 3364 2688 S 0.0 0.1 0:00.08 /usr/sbin/rsyslogd -n
1777 syslog 20 0 250M 3364 2688 S 0.0 0.1 0:00.01 /usr/sbin/rsyslogd -n
1744 syslog 20 0 250M 3364 2688 S 0.0 0.1 0:00.17 /usr/sbin/rsyslogd -n
1778 root 20 0 329M 10736 7016 S 0.0 0.2 0:00.00 /usr/sbin/ModemManager
1782 root 20 0 329M 10736 7016 S 0.0 0.2 0:00.00 /usr/sbin/ModemManager
1757 root 20 0 329M 10736 7016 S 0.0 0.2 0:00.04 /usr/sbin/ModemManager
1764 root 20 0 4400 1352 1256 S 0.0 0.0 0:01.26 /usr/sbin/acpid
1767 lessagebu 20 0 44412 5384 3604 S 0.0 0.1 0:02.85 /usr/bin/dbus-daemon --system --address=systemd: --r
1790 root 20 0 37296 3192 2904 S 0.0 0.1 0:00.09 /usr/sbin/cron -f
1828 root 20 0 292M 10736 7724 S 0.0 0.2 0:01.24 /usr/lib/accountsservice/accounts-daemon
1830 root 20 0 292M 10736 7724 S 0.0 0.2 0:00.02 /usr/lib/accountsservice/accounts-daemon
1793 root 20 0 292M 10736 7724 S 0.0 0.2 0:01.35 /usr/lib/accountsservice/accounts-daemon
1796 root 20 0 28628 3192 2812 S 0.0 0.1 0:00.25 /lib/systemd/systemd-logind
1980 whoopsie 20 0 451M 16288 13008 S 0.0 0.3 0:00.00 /usr/bin/whoopsie -f
1981 whoopsie 20 0 451M 16288 13008 S 0.0 0.3 0:00.02 /usr/bin/whoopsie -f
1798 whoopsie 20 0 451M 16288 13008 S 0.0 0.3 0:00.27 /usr/bin/whoopsie -f
1960 root 20 0 453M 17172 14408 S 0.0 0.3 0:00.08 /usr/sbin/NetworkManager --no-daemon
1962 root 20 0 453M 17172 14408 S 0.0 0.3 0:00.13 /usr/sbin/NetworkManager --no-daemon
1799 root 20 0 453M 17172 14408 S 0.0 0.3 0:01.04 /usr/sbin/NetworkManager --no-daemon
1800 avahi 20 0 44920 3156 2824 S 0.0 0.1 0:00.75 avahi-daemon: running [MichałczukM-VirtualBox.local]
1893 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.33 /usr/lib/snapd/snapd
1896 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.00 /usr/lib/snapd/snapd
1902 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.01 /usr/lib/snapd/snapd
1990 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.45 /usr/lib/snapd/snapd
1991 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.18 /usr/lib/snapd/snapd
1992 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.61 /usr/lib/snapd/snapd
5424 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.20 /usr/lib/snapd/snapd
13962 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.26 /usr/lib/snapd/snapd
13963 root 20 0 233M 15452 9940 S 0.0 0.3 0:00.28 /usr/lib/snapd/snapd
1814 root 20 0 233M 15452 9940 S 0.0 0.3 0:02.44 /usr/lib/snapd/snapd
F1 Del F2 Setup F3 Search F4 Filter F5 Tree F6 Sort By F7 Nice F8 Nice + F9 Kill F10 quit

```

# [Console] Strike 8 – procesy, grep i kill

0) Włącz program *gedit*

1) Zamknij go, korzystając z polecenia `kill`

Wykorzystaj: ps i grep do znalezienia PID.

2) Policz ilość wystąpień słowa ‘**tofu**’ w pliku  
`long-file.txt`

3) Policz ile procesów ma uruchomionych:

- root
- ty (twoja nazwa usera)
- łącznie

**Hint:** Mieliśmy już program do liczenia linii



Dziękuję za uwagę



**michalczukm**

**michalczukm@gmail.com**