



Linux

Trener: Michał Michalczuk

Gdańsk, 23 października 2016 roku

www.infoshareacademy.com

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
- Przydatne narzędzia

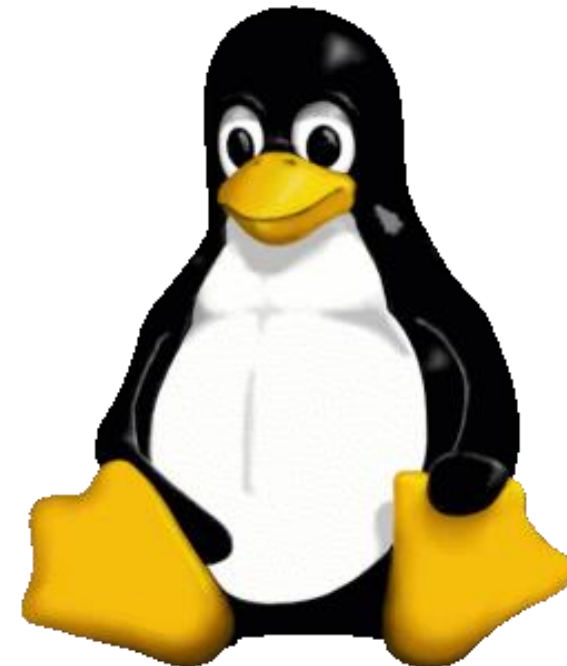
Czym Linux?

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

Unix?

Jądro Linux?

FOSS?



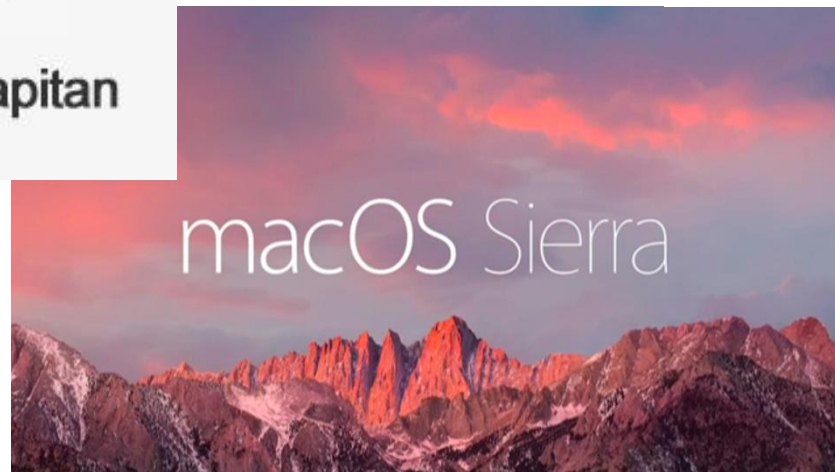
Unix

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

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

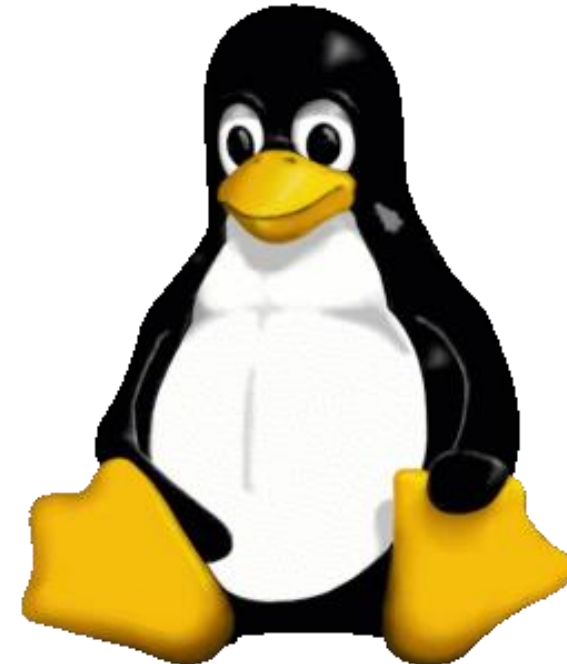


Popularne UNIX-y



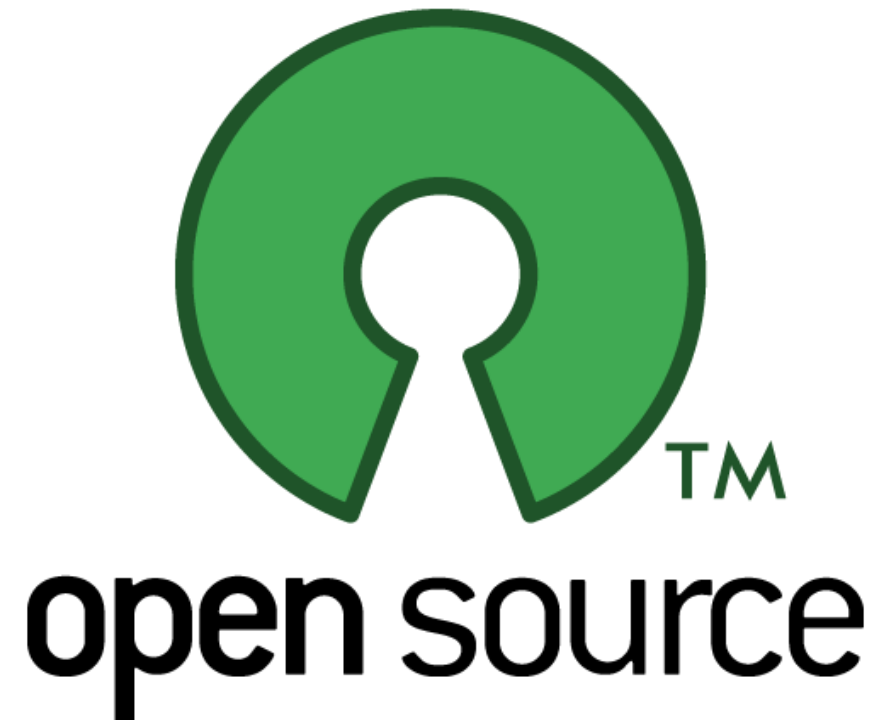
Jądro (ang. *kernel*) Linux

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.



Free and open-source software (FOSS/FLOSS)

Kod źródłowy może być dowolnie wykorzystywany, modyfikowany i rozpowszechniany.



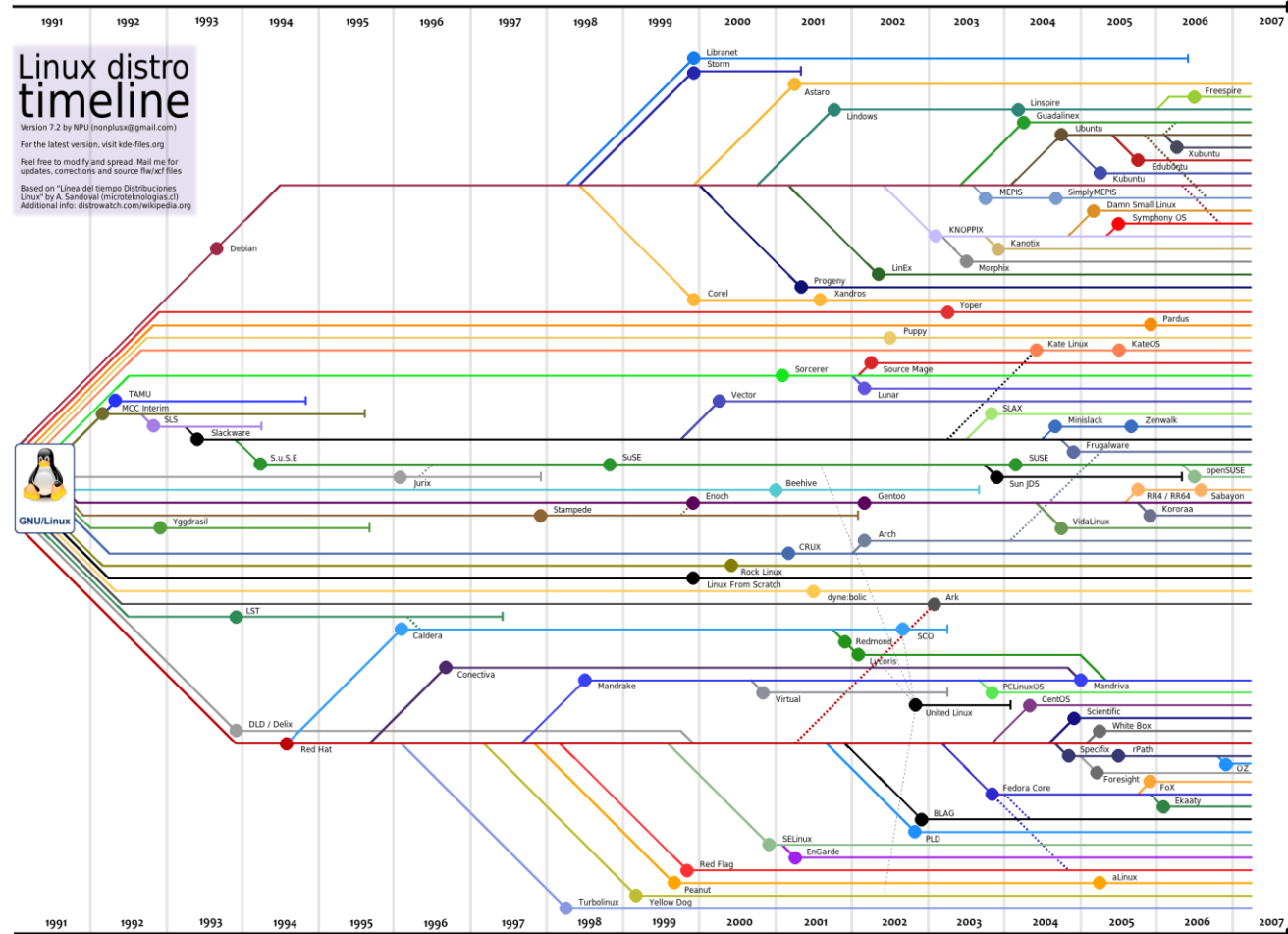
Linus Torvalds

- 1991 Linux kernel
- 2005 Git

Linux = Linus + Unix



Dystrybucje Linux



Dystrybucje Linux



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

Pictures

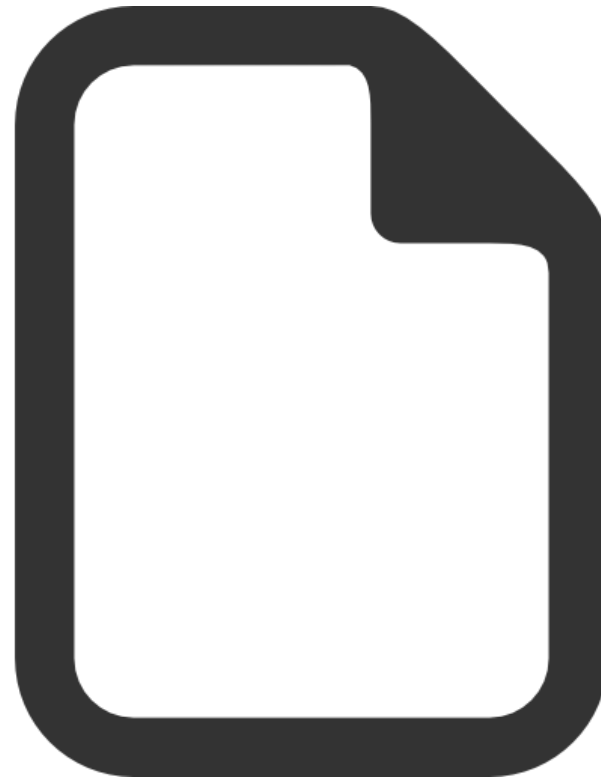
[Exercise] Proszę się nie bać Linuxa



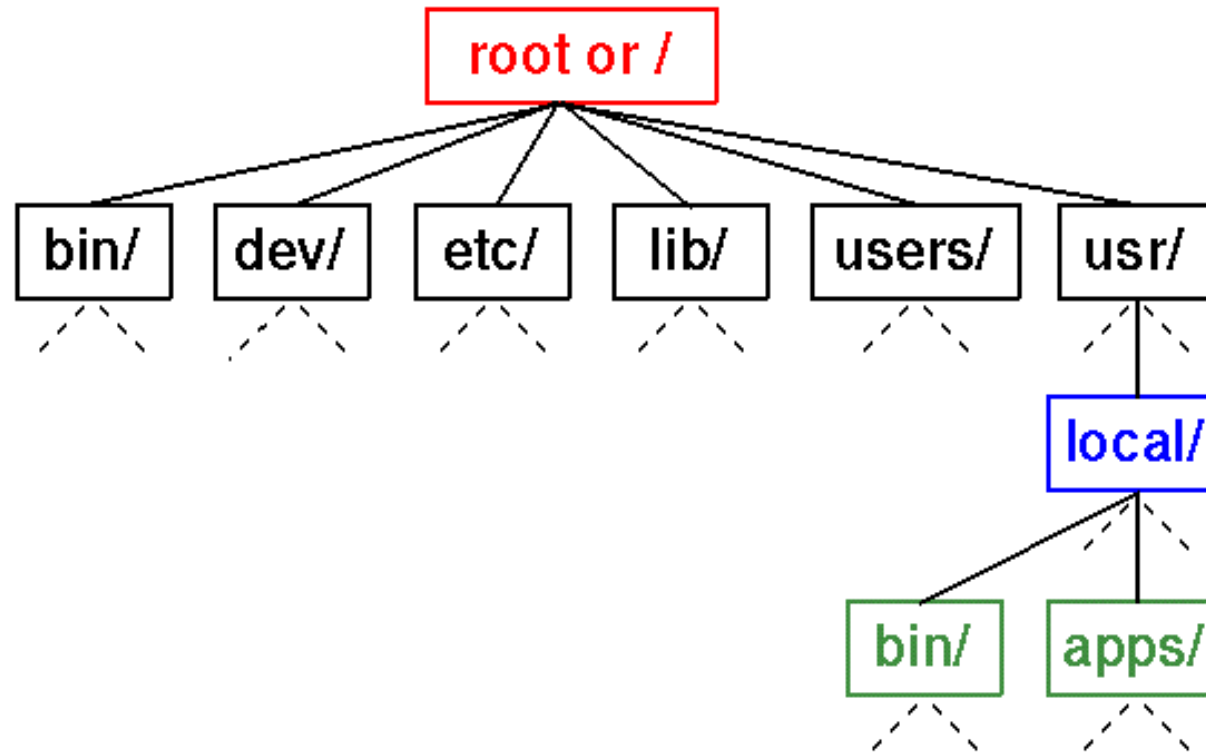
Struktura folderów i katalogów



Wszystko jest plikiem ... wszystko



FHS (File Hierarchy Standard)



Więcej: [struktura katalogów](#)

Podstawowa struktura katalogów

/ - root

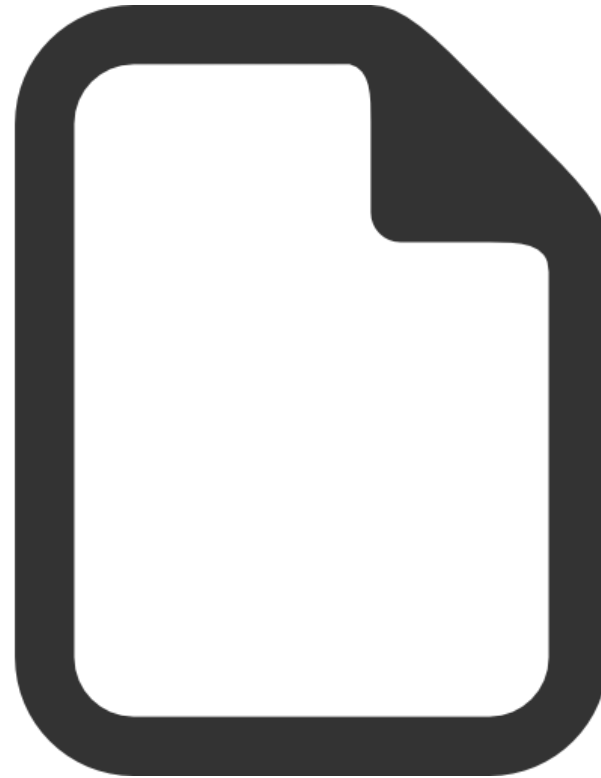
/home - katalogi domowe użytkowników

/root - katalog domowy administratora

/usr - programy użytkowników

/etc - ustawienia systemowe .. taki panel sterowania

Wszystko jest plikiem ... wszystko



Podstawowa struktura katalogów

/dev - urządzenia, pseudopliki o specjalnych uprawnieniach

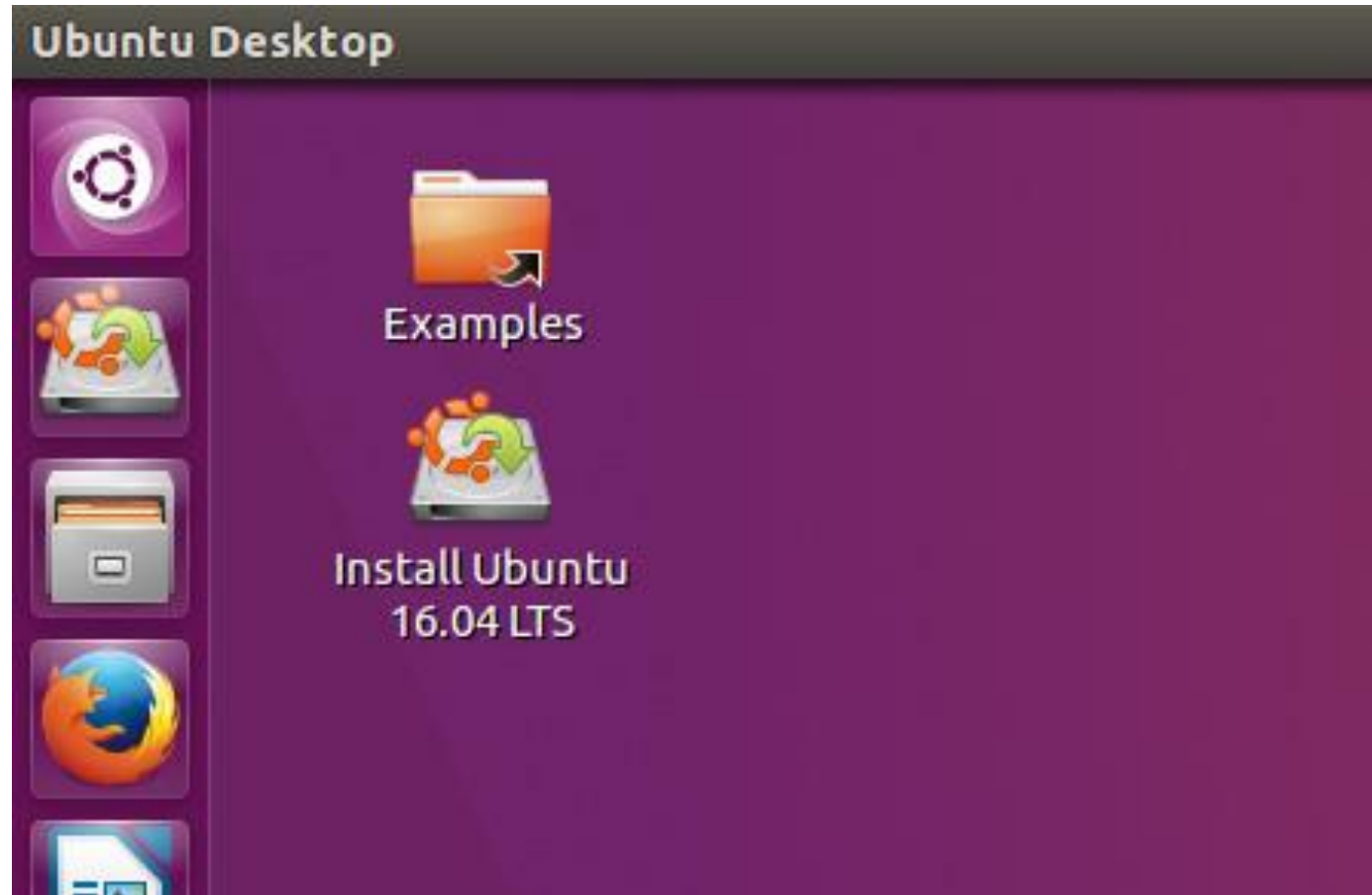
/media - zamontowane dyski, napęd CD, pendrive

/var- logi systemowe i logi wszelakie

/proc - dane o zasobach i aktualnie 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

ll {path} - pokaż listę plików

ll .. - pokaż listę plików katalog wyżej

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

ll ~ - pokaż do twojego katalogu domowego

np.:

ll Pictures/

ll ~/Pictures/

Output ll – 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 ll – 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

[Console] Strike 1 – poruszanie się po katalogach



Parę przydatnych komend

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

Parę przydatnych komend

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

[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 -lsh {path}

Parametry:

- l
- S
- h

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**.

Głównie parametry i ich znaczenie.

Zależy od dobrej woli programisty :) To tylko konwencja.

[Console] Strike 2 – pomoc



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

rwX

rwX

user

group

others

Kiedy mamy uprawnienie?

0 – brak uprawnienia

1 – jest uprawnienie

r w x ----> 111

- - - ----> 000

r - x ----> 101

- - x ----> 001

Kiedy mamy uprawnienie?

Przykłady:

d rwx r-x --x my_temp_dir

- r-x -w- --- my-note.txt

- rwx r-x r-x my_script.sh

chmod (*change mode*)

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

Jest tutaj parę możliwości.

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/atryb.htm>

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

[Console] Strike 4 – uprawnienia

~/linux_classes

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



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-get install – instalacja pakietu

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

np.:

apt-get install tree

apt-get 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.



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 przydatke komenty 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),

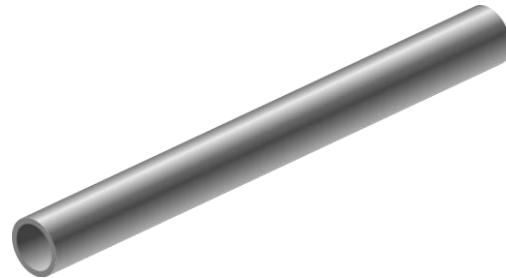
oraz przetestuj (uruchom) pakiety:

- **guake**
- tree
- mc (to inaczej program mcedit)

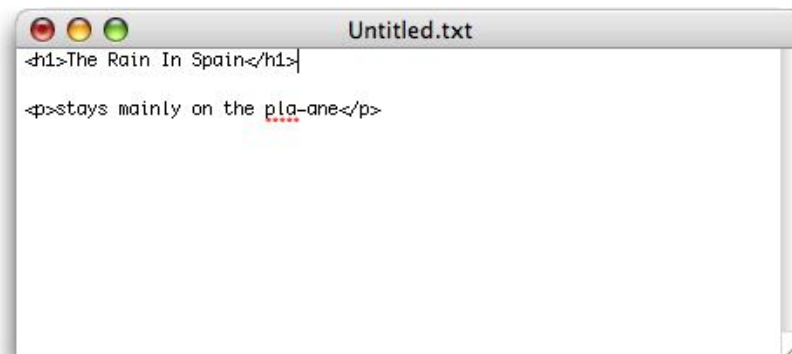
2. Usuń (ale wyłącznie z konfiguracją i "wyczyść" potem apt-get) pakiet **guake**



Pipes | potoki

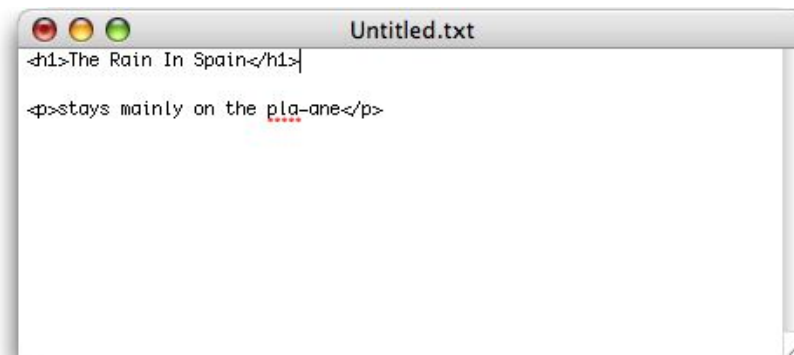


Edytory tekstu: mcedit, nano, vim



```
<h1>The Rain In Spain</h1>|  
<p>stays mainly on the pla-ane</p>
```

Tools: tar, zip, grep



```
<h1>The Rain In Spain</h1>
<p>stays mainly on the pla-ane</p>
```

Dziękuję za uwagę



michalczukm



michalczukm@gmail.com