

Podstawy informatyki

WYKŁAD nr 01

Fizyka Techniczna, WFT PP

Michał Hermanowicz

Zakład Fizyki Obliczeniowej i Nanomechaniki, Instytut Fizyki, Politechnika Poznańska

Rok akademicki 2018/2019



1 Organizacja

- Cel i treść przedmiotu
- Forma zajęć i **warunki zaliczenia**
- Materiały dydaktyczne
- Literatura

2 Plan ramowy przedmiotu

3 Wprowadzenie

- System Unix (AT&T Bell Labs)
- Projekt GNU
- System operacyjny GNU/Linux
- Licencje na oprogramowanie komputerowe
- System plików
- Interfejs użytkownika

4 Podsumowanie: pytania i dyskusja

Cel i treść przedmiotu

Podstawy informatyki:

przedmiot ma charakter **narzędziowy** i stanowi przygotowanie do zajęć odbywających się w kolejnych semestrach I i II stopnia kształcenia.

Cel i treść przedmiotu

Podstawy informatyki:

przedmiot ma charakter **narzędziowy** i stanowi przygotowanie do zajęć odbywających się w kolejnych semestrach I i II stopnia kształcenia.

Poruszane zagadnienia:

- system operacyjny GNU/Linux i jego obsługa,
- powłoka BASH i jej podstawowe instrukcje,
- podstawowe programy użytkowe systemu,
- przetwarzanie zbiorów danych, ich formatowanie i metody graficznej prezentacji na wykresach 2D i 3D,
- elementy programowania w środowisku GNU/Linux,
- tworzenie dokumentów w systemie składu tekstu \LaTeX .

Forma zajęć i warunki zaliczenia

- 1 Wykład (sala D):
 - ▶ wymiar: 15h,
 - ▶ **zaliczenie**: kolokwium pisemne na ostatnim wykładzie.

Forma zajęć i warunki zaliczenia

❶ Wykład (sala D):

- ▶ wymiar: 15h,
- ▶ **zaliczenie**: kolokwium pisemne na ostatnim wykładzie.

❷ Pracownia (sala 602 (601)):

- ▶ wymiar: 10h + 5h,
- ▶ praktyczne przećwiczenie zagadnień poruszanych na wykładzie,
- ▶ **zaliczenie**: sprawdzian umiejętności z wykorzystaniem komputera i oprogramowania (na ostatnich zajęciach); warunkiem przystąpienia do zaliczenia jest obecność i czynny udział w zajęciach,
- ▶ nieobecności:
 - ★ zwolnienie lekarskie (do 2 tygodni po nieobecności),
 - ★ nie mogą przekroczyć 1/3 wszystkich zajęć,
 - ★ regulamin studiów PP (z dn. 25.03.2015).
- ▶ prowadzący zajęcia w pracowni:
 - ★ dr inż. A.A. Poźniak (ZFOiN, IF PP) – grupy: 1-3 (10h),
 - ★ dr inż. M. Hermanowicz (ZFOiN, IF PP) – grupy: 4-6 (10h),
 - ★ dr inż. M. Nowicki (ZFPiN, IF PP) – grupy: 1-6 (5h).

Forma zajęć i warunki zaliczenia

Stosujemy klasyczną skalę ocen. Ocenę pozytywną zapewnia przynajmniej 50% poprawnych odpowiedzi z kolokwium/sprawdzianu zaliczeniowego.

W przypadku zajęć w pracowni maksymalna liczba punktów jest sumą:

- I część (10h): 10 punktów,
- II część (5h): 5 punktów.

Konwersja punktów na oceny:

< 50%	2.0
50-60%	3.0
61-70%	3.5
71-80%	4.0
81-90%	4.5
91-100%	5.0

Źródło informacji i materiałów:

<http://www.hermanowicz.eu>

Materiały dydaktyczne będą udostępniane w trakcie semestru.

Podstawa:

- materiały dydaktyczne udostępniane w ramach zajęć;
- dokumentacja wykorzystywanego oprogramowania dostępna *on-line* oraz w pracowni komputerowej.

Uzupełnienie:

- 101 skryptów w shellu; Dave Taylor; Mikom, W-wa, 2004;
- [Bash. Receptury](#); C. Albing, J.P. Vossen, C. Newham; Helion, 2008;
- [Organizacja i architektura systemu komputerowego: projektowanie systemu a jego wydajność](#); W. Stallings; WNT, W-wa, 2003;
- [Język ANSI C](#); B.W. Kernighan, D.M. Ritchie; WNT, W-wa, 2004;
- dodatkowe (specjalistyczne) pozycje będą podawane w trakcie wykładów/zajęć w pracowni.

Plan ramowy przedmiotu

Nr wykładu	Poruszane zagadnienia
I	Organizacja; forma i warunki zaliczenia; wprowadzenie
II	Powłoka bash i elementy programowania
III	Przetwarzanie danych #1
IV	Przetwarzanie danych #2
V	Reprezentacja danych (wykresy 2D i 3D) – <i>gnuplot</i>
VI	System składu tekstu \LaTeX
VII	Pół-otwarty test zaliczeniowy

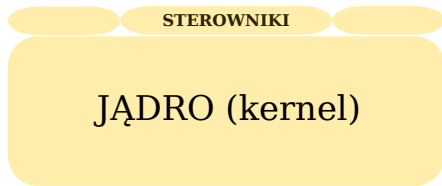
Każdemu z wykładów odpowiadają ćwiczenia realizowane na zajęciach w pracowni komputerowej.

Co to jest system operacyjny?

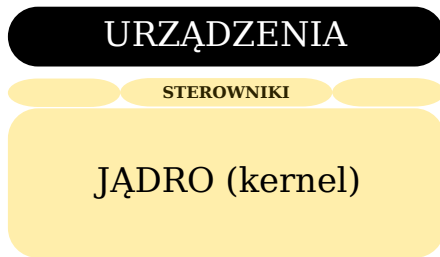
Co to jest system operacyjny?

JĄDRO (kernel)

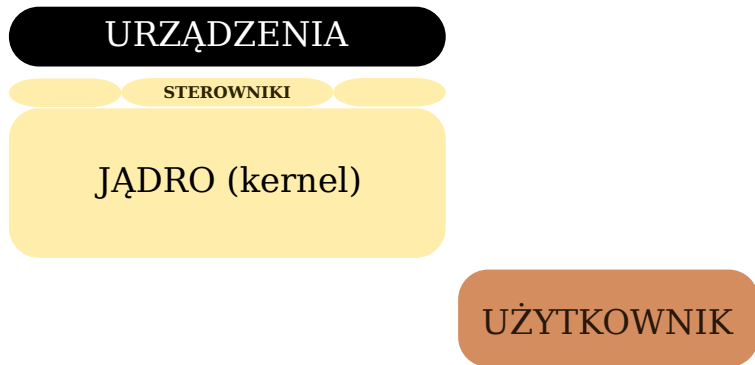
Co to jest system operacyjny?



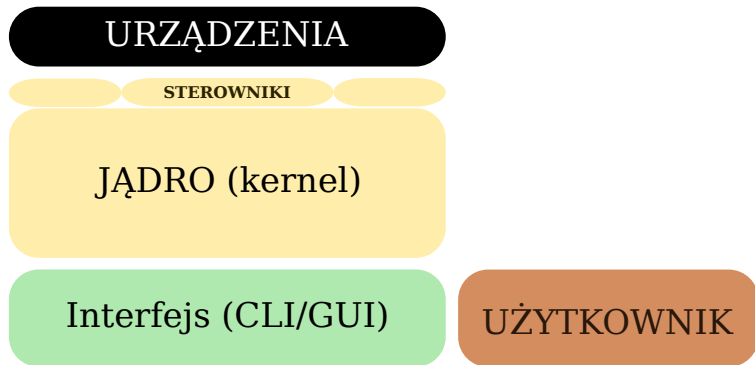
Co to jest system operacyjny?



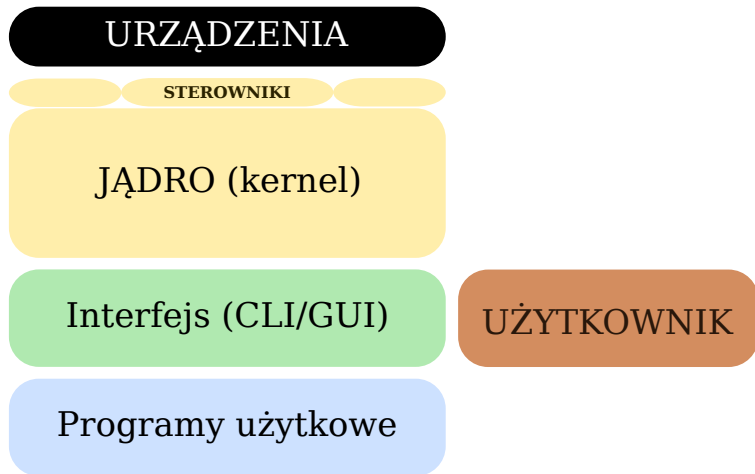
Co to jest system operacyjny?



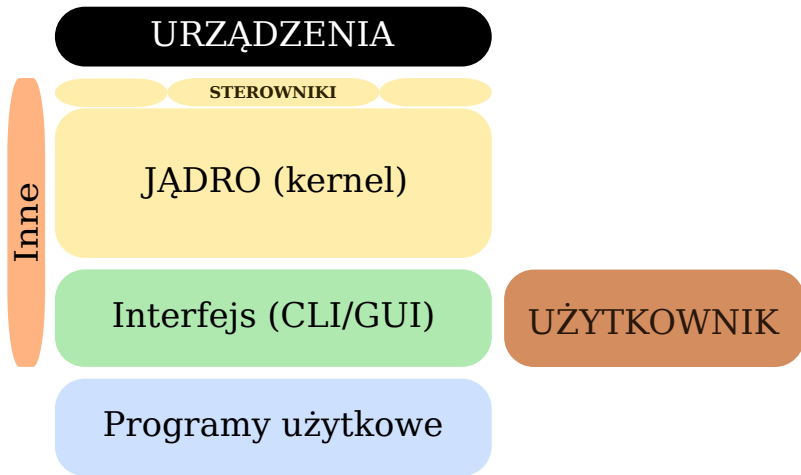
Co to jest system operacyjny?



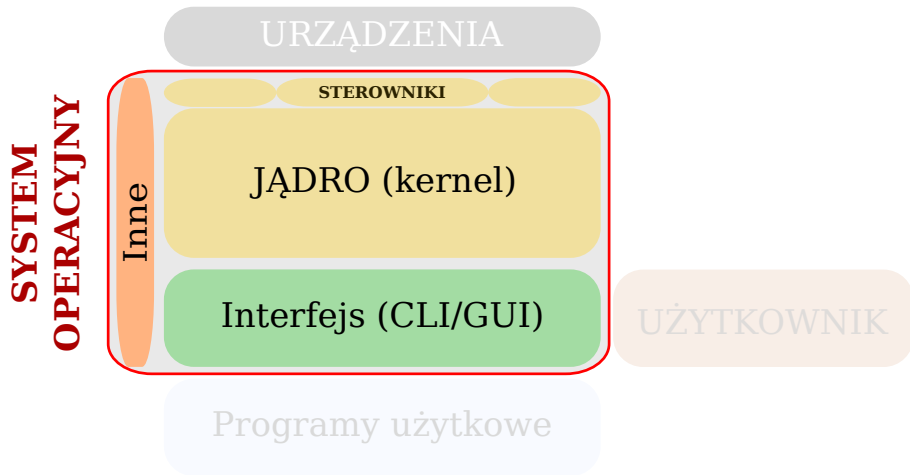
Co to jest system operacyjny?



Co to jest system operacyjny?



Co to jest system operacyjny?



Ken Thompson i Dennis Ritchie (AT&T Bell Labs)



Zdjęcie: **Peter Hamer** [CC BY-SA 2.0 (<http://creativecommons.org/licenses/by-sa/2.0>)],
via Wikimedia Commons

System operacyjny Unix: filozofia i cechy

Filozofia (autor: Doug McIlroy)¹:

- *write programs that do **one thing** and do it **well**,*
- *write programs to **work together**,*
- *write programs that handle **text** streams as **a universal interface**.*

¹Cytowane punkty pochodzą z: **Peter H. Salus**, *A Quarter-Century of Unix*.

Addison-Wesley, 1994. ISBN 0-201-54777-5.

System operacyjny Unix: filozofia i cechy

Filozofia (autor: Doug McIlroy)¹:

- *write programs that do **one thing** and do it **well**,*
- *write programs to **work together**,*
- *write programs that handle **text** streams as **a universal interface**.*

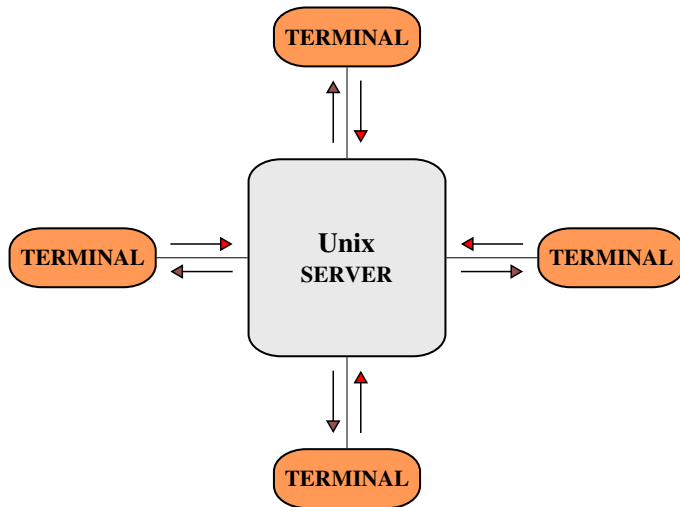
¹Cytowane punkty pochodzą z: **Peter H. Salus**, *A Quarter-Century of Unix*.

Addison-Wesley, 1994. ISBN 0-201-54777-5.

Cechy systemu Unix:

- wielozadaniowy, wieloużytkowy system operacyjny (*timesharing* OS),
- monolityczne jądro, hierarchiczny system plików,
- *wszystko jest plikiem* (nawet urządzenia!),
- obsługa sieci, napisany (przepisany) w języku wysokiego poziomu,
- koncepcja **pipeline** (*potok*).

System operacyjny Unix: środowisko wieloużytkowe



System operacyjny Unix: terminal VT100



Zdjęcie: **Jason Scott** [CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0>)],
via Wikimedia Commons

Problem z Uniksem:

Problem z Uniksem:
ograniczenia licencyjne!

Projekt GNU: Richard M. Stallman

- 1983 – zapowiedź nowego systemu operacyjnego, który będzie **JAK Unix** (z jego filozofią i cechami), ale całkowicie **wolny**,
- licencja typu **copyleft**,
- publikacja manifestu GNU,
- GNU = “**GNU’s Not Unix**” (www.gnu.org),
- *Free Software Foundation* (www.fsf.org).



Zdjęcie: artykuł **Daniele Piccinelli** (2016), <https://www.macitynet.it>



- Brakujący (istotny) element systemu!



- Brakujący (istotny) element systemu!
- Jądro **Hurd** na zbyt wczesnym etapie rozwoju. . .

GNU Hurd

[Recent Changes](#)[Preferences](#)*This page:*[Edit](#)[History](#)[Source](#)[?Discussion](#)

What is the GNU Hurd?

The GNU Hurd is the GNU project's replacement for the Unix kernel. It is a collection of servers that run on the Mach microkernel to implement file systems, network protocols, file access control, and other features that are implemented by the Unix kernel or similar kernels (such as Linux). [More detailed.](#)

What is the mission of the GNU Hurd project?

Our mission is to create a general-purpose kernel suitable for the GNU operating system, which is viable for everyday use, and gives users and programs as much control over their computing environment as possible. [Our mission explained.](#)

Screenshot: strona domowa projektu GNU Hurd (<https://www.gnu.org/software/hurd>).

Date: 25 August 1991

From: torvalds@kruuna.helsinki.fi

Hello everybody out there using minix --

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things). (...)

Linus (torvalds@kruuna.helsinki.fi)





= GNU/Linux



= **GNU/Linux**

- GNU/Linux to nie jest jeden system operacyjny – istnieje wiele jego **dystrybucji**, z których każda posiada swoją specyfikę.



= **GNU/Linux**

- GNU/Linux to nie jest jeden system operacyjny – istnieje wiele jego **dystrybucji**, z których każda posiada swoją specyfikę.
- Dostępny dla wielu architektur sprzętowych: amd64, arm64, i386, powerpc, mips, s390x i innych.



= **GNU/Linux**

- GNU/Linux to nie jest jeden system operacyjny – istnieje wiele jego **dystrybucji**, z których każda posiada swoją specyfikę.
- Dostępny dla wielu architektur sprzętowych: amd64, arm64, i386, powerpc, mips, s390x i innych.
- GNU: www.gnu.org / Linux: www.kernel.org

Niektóre z dystrybucji GNU/Linuxu:

- **Debian** [www.debian.org]
- **Slackware Linux** [www.slackware.com]
- **Arch Linux** [www.archlinux.org]
- **PLD** [www.pld-linux.org]
- **gNewSense** [www.gnewsense.org]
- **Trisquel** [www.trisquel.info]
- **Fedora** [www.getfedora.org]
- **CentOS** [www.centos.org]
- **Scientific Linux** [www.scientificlinux.org]



Logotypy: **GNU** (na licencji GFDL 1.3, źródło: www.gnu.org);
Linux (Tux): Larry Ewing, lewing@isc.tamu.edu, GIMP (www.gimp.org).

KOD ŹRÓDŁOWY

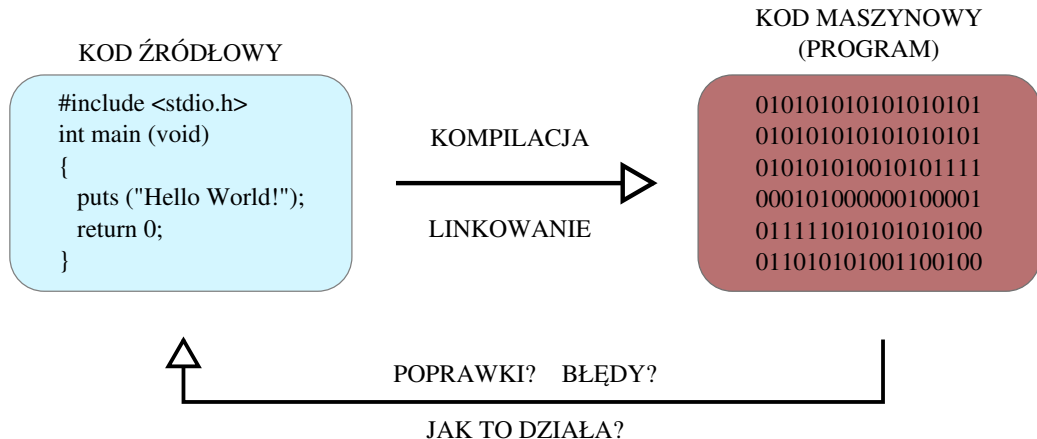
```
#include <stdio.h>
int main (void)
{
    puts ("Hello World!");
    return 0;
}
```

KOMPILACJA

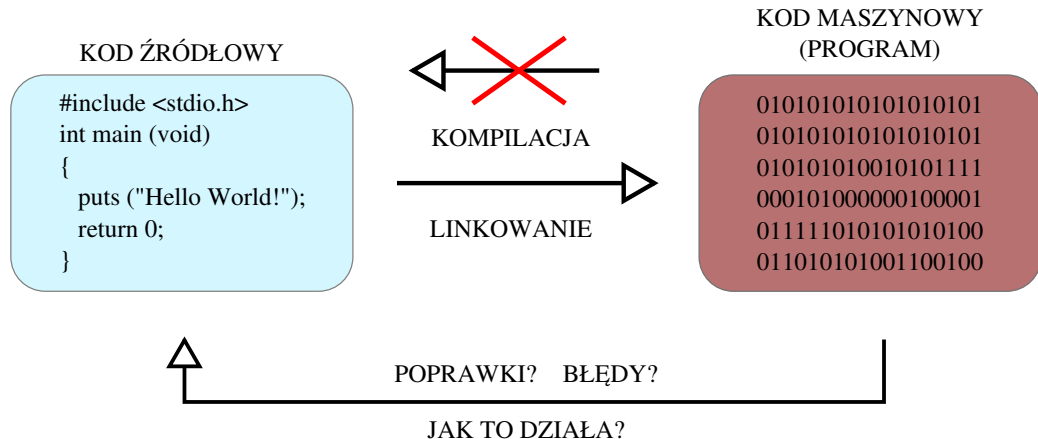
LINKOWANIE

KOD MASZYNOWY (PROGRAM)

```
010101010101010101
010101010101010101
010101010010101111
000101000000100001
011111010101010100
011010101001100100
```



Licencje na oprogramowanie komputerowe



Wolne oprogramowanie (WO) = cztery fundamentalne **wolności**:

Wolne oprogramowanie (WO) = cztery fundamentalne **wolności**:

- I do uruchamiania programu w dowolnym celu i w dowolny sposób;

Wolne oprogramowanie (WO) = cztery fundamentalne **wolności**:

- I do uruchamiania programu w dowolnym celu i w dowolny sposób;
- II do studiowania kodu źródłowego (*open source*);

Wolne oprogramowanie (WO) = cztery fundamentalne **wolności**:

- I do uruchamiania programu w dowolnym celu i w dowolny sposób;
- II do studiowania kodu źródłowego (*open source*);
- III do rozpowszechniania kopii programu w dowolny sposób;

Wolne oprogramowanie (WO) = cztery fundamentalne **wolności**:

- I do uruchamiania programu w dowolnym celu i w dowolny sposób;
- II do studiowania kodu źródłowego (*open source*);
- III do rozpowszechniania kopii programu w dowolny sposób;
- IV do modyfikowania programu i rozpowszechniania zmienionych wersji.

Wolne oprogramowanie (WO) = cztery fundamentalne **wolności**:

- I do uruchamiania programu w dowolnym celu i w dowolny sposób;
- II do studiowania kodu źródłowego (*open source*);
- III do rozpowszechniania kopii programu w dowolny sposób;
- IV do modyfikowania programu i rozpowszechniania zmienionych wersji.

Czego zrobić **nie wolno**:

Wolne oprogramowanie (WO) = cztery fundamentalne **wolności**:

- I do uruchamiania programu w dowolnym celu i w dowolny sposób;
- II do studiowania kodu źródłowego (*open source*);
- III do rozpowszechniania kopii programu w dowolny sposób;
- IV do modyfikowania programu i rozpowszechniania zmienionych wersji.

Czego zrobić nie wolno:

jeżeli rozpowszechniamy wolne oprogramowanie, to nie wolno nam odebrać innym użytkownikom żadnej z powyższych czterech wolności.

Licencje na oprogramowanie komputerowe

Wolne oprogramowanie (WO) = cztery fundamentalne **wolności**:

- I do uruchamiania programu w dowolnym celu i w dowolny sposób;
- II do studiowania kodu źródłowego (*open source*);
- III do rozpowszechniania kopii programu w dowolny sposób;
- IV do modyfikowania programu i rozpowszechniania zmienionych wersji.

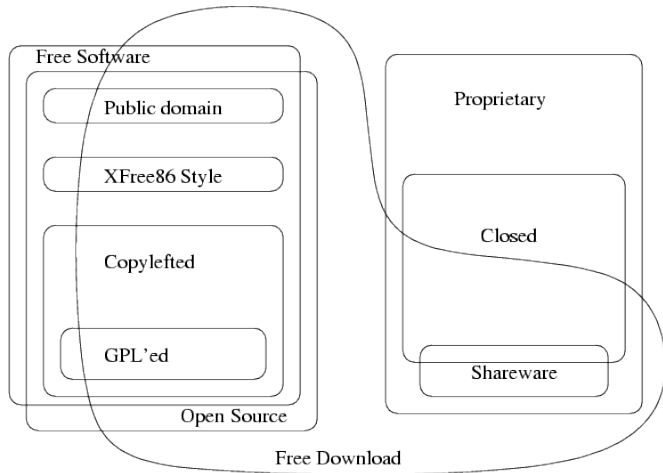
Czego zrobić **nie wolno**:

jeżeli rozpowszechniamy wolne oprogramowanie, to nie wolno nam odebrać innym użytkownikom żadnej z powyższych czterech wolności.

W praktyce:

GNU General Public License 3.0 (<https://www.gnu.org/licenses/gpl-3.0.txt>) lub jedna z około 100 innych! Lista wybranych licencji: <https://www.gnu.org/licenses/>

Licencje na oprogramowanie komputerowe



Rysunek: Copyright (c) 2003-2005 Francesco Potorti. *Verbatim copying and distribution of this entire article is permitted in any medium, provided that this notice is preserved.*

Dlaczego to jest ważne:

- sposób uprawiania nauki jest kompatybilny z modelem rozwoju wolnego oprogramowania,

Dlaczego to jest ważne:

- sposób uprawiania nauki jest kompatybilny z modelem rozwoju wolnego oprogramowania,
- swobodny przepływ myśli technicznej i algorytmów (tak funkcjonuje środowisko akademickie),

Dlaczego to jest ważne:

- sposób uprawiania nauki jest kompatybilny z modelem rozwoju wolnego oprogramowania,
- swobodny przepływ myśli technicznej i algorytmów (tak funkcjonuje środowisko akademickie),
- dobra podstawa do tworzenia solidnej dokumentacji programów,

Dlaczego to jest ważne:

- sposób uprawiania nauki jest kompatybilny z modelem rozwoju wolnego oprogramowania,
- swobodny przepływ myśli technicznej i algorytmów (tak funkcjonuje środowisko akademickie),
- dobra podstawa do tworzenia solidnej dokumentacji programów,
- w zastosowaniach naukowych **musimy** wiedzieć jak dokładnie działa program – z jakich rozwiązań i algorytmów korzysta,

Dlaczego to jest ważne:

- sposób uprawiania nauki jest kompatybilny z modelem rozwoju wolnego oprogramowania,
- swobodny przepływ myśli technicznej i algorytmów (tak funkcjonuje środowisko akademickie),
- dobra podstawa do tworzenia solidnej dokumentacji programów,
- w zastosowaniach naukowych **musimy** wiedzieć jak dokładnie działa program – z jakich rozwiązań i algorytmów korzysta,
- dostępność kodu źródłowego umożliwia swobodną edukację,

Dlaczego to jest ważne:

- sposób uprawiania nauki jest kompatybilny z modelem rozwoju wolnego oprogramowania,
- swobodny przepływ myśli technicznej i algorytmów (tak funkcjonuje środowisko akademickie),
- dobra podstawa do tworzenia solidnej dokumentacji programów,
- w zastosowaniach naukowych **musimy** wiedzieć jak dokładnie działa program – z jakich rozwiązań i algorytmów korzysta,
- dostępność kodu źródłowego umożliwia swobodną edukację,
- szybki rozwój oprogramowania – duża ilość poprawek pochodzących od niezależnych programistów,

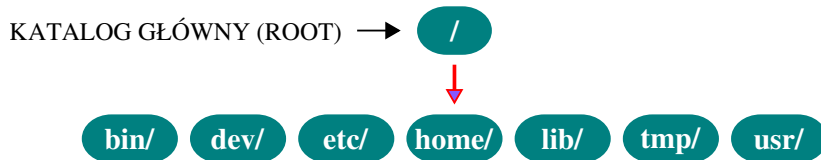
Dlaczego to jest ważne:

- sposób uprawiania nauki jest kompatybilny z modelem rozwoju wolnego oprogramowania,
- swobodny przepływ myśli technicznej i algorytmów (tak funkcjonuje środowisko akademickie),
- dobra podstawa do tworzenia solidnej dokumentacji programów,
- w zastosowaniach naukowych **musimy** wiedzieć jak dokładnie działa program – z jakich rozwiązań i algorytmów korzysta,
- dostępność kodu źródłowego umożliwia swobodną edukację,
- szybki rozwój oprogramowania – duża ilość poprawek pochodzących od niezależnych programistów,
- aspekty etyczne – wśród nich brak rozwiązań typu **DRM** i potencjalnie *złośliwej* (*malware*) funkcjonalności (niektórych) programów własnościowych.

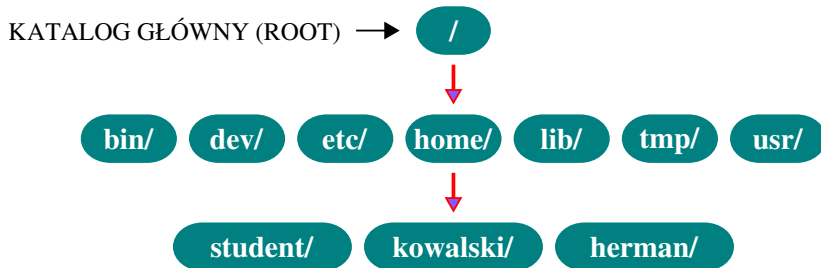
Rzecz o systemie plików

KATALOG GŁÓWNY (ROOT) → 

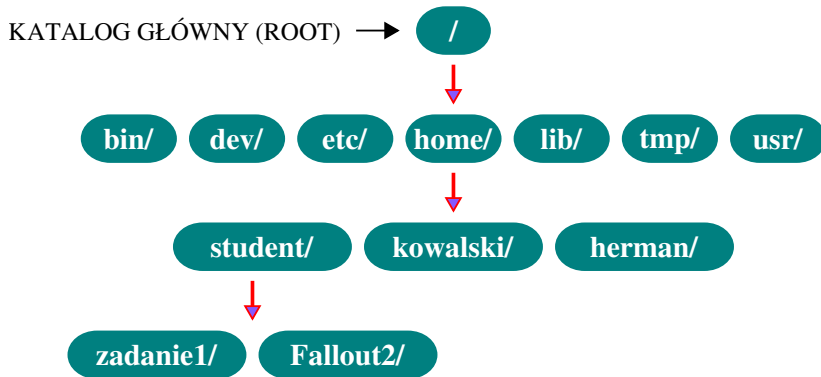
System operacyjny GNU/Linux: **system plików**



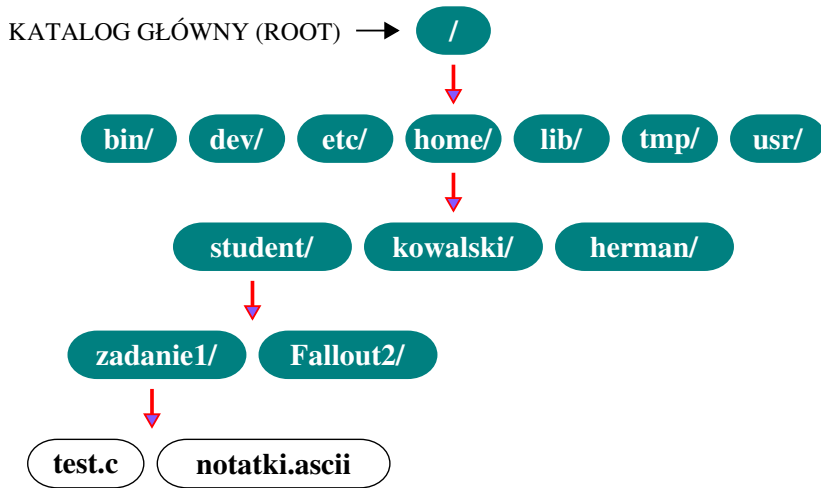
System operacyjny GNU/Linux: **system plików**



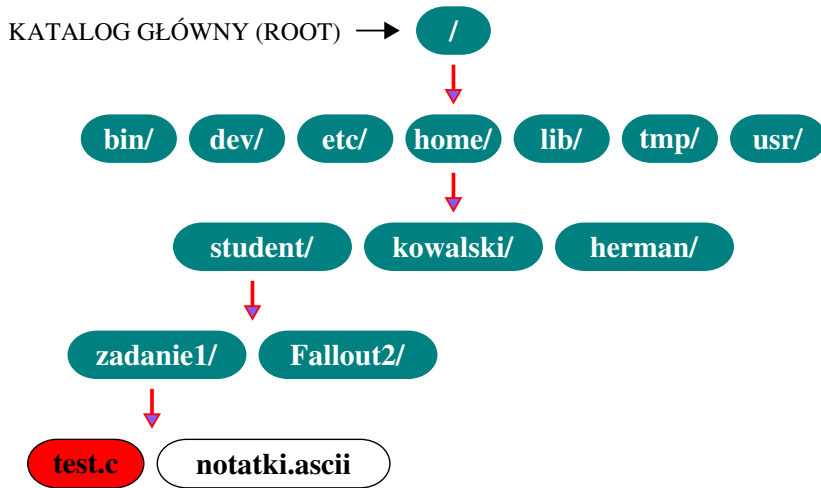
System operacyjny GNU/Linux: **system plików**



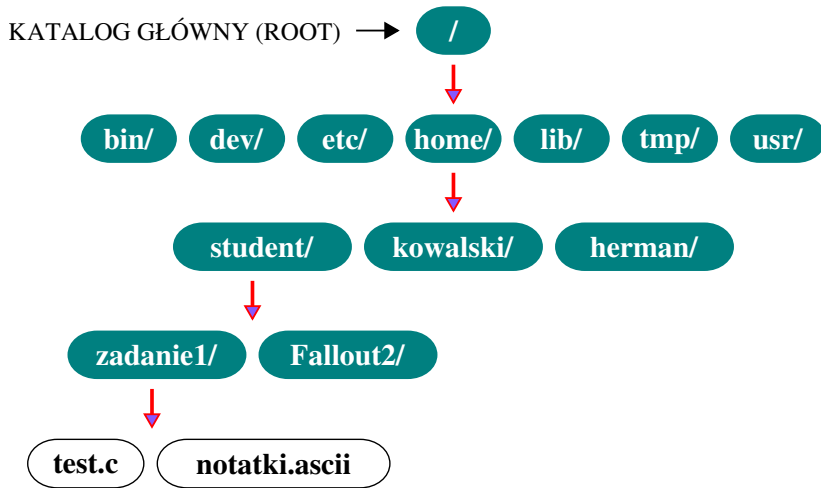
System operacyjny GNU/Linux: **system plików**



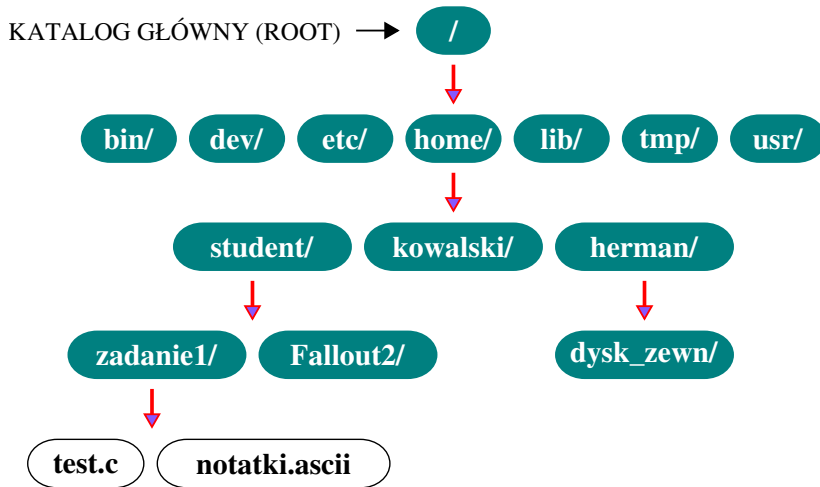
System operacyjny GNU/Linux: system plików



System operacyjny GNU/Linux: **system plików**



System operacyjny GNU/Linux: **system plików**



System operacyjny GNU/Linux: **system plików**

Bezwzględna ścieżka dostępu:

```
/home/student/zadanie1/test.c
```

System operacyjny GNU/Linux: **system plików**

Bezwzględna ścieżka dostępu:

`/home/student/zadanie1/test.c`

Względna ścieżka dostępu:

`zadanie1/test.c`

System operacyjny GNU/Linux: system plików

Bezwzględna ścieżka dostępu:

`/home/student/zadanie1/test.c`

Względna ścieżka dostępu:

`zadanie1/test.c`

Ważne katalogi:

- `/bin` – pliki wykonywalne (**binary**), programy użytkowe i inne,
- `/dev` – pliki urządzeń (dyski, drukarki i inne),
- `/etc` – pliki konfiguracyjne systemu,
- `/home` – katalogi domowe użytkowników,
- `/lib` – biblioteki systemowe,
- `/tmp` – pliki tymczasowe,
- `/usr` – oprogramowanie użytkownika wraz z bibliotekami, dok. i in.

W nazwach plików i katalogów:

- **NIE** używamy SPACJI*,
- **NIE** używamy znaków diakrytycznych*.

*Choć jest to technicznie możliwe, to znacznie komplikuje wiele czynności i w efekcie utrudnia korzystanie z systemu, dlatego taką zasadę warto przyjąć jako dobrą praktykę.

W nazwach plików i katalogów:

- **NIE** używamy SPACJI*,
- **NIE** używamy znaków diakrytycznych*.

*Choć jest to technicznie możliwe, to znacznie komplikuje wiele czynności i w efekcie utrudnia korzystanie z systemu, dlatego taką zasadę warto przyjąć jako dobrą praktykę.

Domyślnie użytkownik ma uprawnienia do zapisu w katalogach:

- `/home/uzytkownik/`
- `/tmp/`

System operacyjny GNU/Linux: **system plików**

W nazwach plików i katalogów:

- **NIE** używamy SPACJI*,
- **NIE** używamy znaków diakrytycznych*.

*Choć jest to technicznie możliwe, to znacznie komplikuje wiele czynności i w efekcie utrudnia korzystanie z systemu, dlatego taką zasadę warto przyjąć jako dobrą praktykę.

Domyślnie użytkownik ma uprawnienia do zapisu w katalogach:

- `/home/uzytkownik/`
- `/tmp/`

Niektóre z obsługiwanych systemów plików:

EXT2/EXT3/EXT4, ReiserFS, XFS, ZFS, JFS, FAT16/32, NTFS

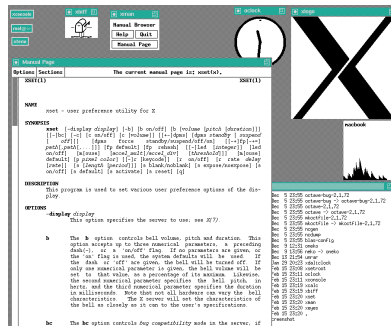
System operacyjny GNU/Linux: interfejs użytkownika

```
cron.weekly
csh
csh.cshrc
csh.login
csh.logout
cups
cupshelpers
dbus-1
debconf.conf
debian_version
default
deluser.conf
dhcp
dictionaries-common
discover.conf.d
discover-modprobe.conf
dkms
dpkg
drirc
herman@lepton:/etc$
herman@lepton:/etc$
herman@lepton:/etc$
```

```
hosts
hosts.allow
hosts.deny
hp
htdig
i3
i3status.conf
icedove
icedtea-web
iceweasel
idmappd.conf
ifplugd
ImageMagick-6
init
init.d
initramfs-tools
inputrc
insserv
insserv.conf
```

CLI

(Command Line Interface)



GUI

(Graphical User Interface)

Zrzut ekranu z prawej: Liberal Classic [MIT (<http://opensource.org/licenses/mit-license.php>)],

via Wikimedia Commons

GNU Bash (**B**ourne **A**gain **S**hell):

interpreter języka poleceń – **powłoka** (*shell*) systemu GNU/Linux¹.

¹ *The GNU Bash Reference Manual*, v. 4.3 (<http://www.gnu.org>).

System operacyjny GNU/Linux: CLI – **Bash**

GNU Bash (**B**ourne **A**gain **S**hell):

interpreter języka poleceń – **powłoka** (*shell*) systemu GNU/Linux¹.

¹ *The GNU Bash Reference Manual*, v. 4.3 (<http://www.gnu.org>).

Ponadto:

- jest domyślną powłoką systemu GNU/Linux – jego **CLI** (Command Line Interface),
- umożliwia pracę interaktywną (wprowadzanie poleceń), a także wsadową (wykonywanie skryptów).

Inne powłoki: sh, csh, tcsh, ksh, zsh.

System operacyjny GNU/Linux: **CLI**

Debian GNU/Linux wftlab-180 tty1

wftlab-180 login:

System operacyjny GNU/Linux: **CLI**

Debian GNU/Linux wftlab-180 tty1

wftlab-180 login: student

System operacyjny GNU/Linux: **CLI**

Debian GNU/Linux wftlab-180 tty1

wftlab-180 login: student

Password:

System operacyjny GNU/Linux: **CLI**

```
Debian GNU/Linux wftlab-180 tty1
```

```
wftlab-180 login: student
```

```
Password:
```

```
student@wftlab-180:~$
```

System operacyjny GNU/Linux: **CLI**

```
Debian GNU/Linux wftlab-180 tty1
```

```
wftlab-180 login: student
```

```
Password:
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$
```

System operacyjny GNU/Linux: **CLI**

```
Debian GNU/Linux wftlab-180 tty1
```

```
wftlab-180 login: student
```

```
Password:
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$
```


System operacyjny GNU/Linux: **CLI**

```
Debian GNU/Linux wftlab-180 tty1
```

```
wftlab-180 login: student
```

```
Password:
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$
```

System operacyjny GNU/Linux: **CLI**

Debian GNU/Linux wftlab-180 tty1

wftlab-180 login: student

Password:

student@wftlab-180:~\$

student@wftlab-180:~\$

student@wftlab-180:~\$

student@wftlab-180:~\$ pwd

System operacyjny GNU/Linux: **CLI**

```
Debian GNU/Linux wftlab-180 tty1
```

```
wftlab-180 login: student
```

```
Password:
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$ pwd
```

```
/home/student
```

```
student@wftlab-180:~$
```

System operacyjny GNU/Linux: **CLI**

```
Debian GNU/Linux wftlab-180 tty1
```

```
wftlab-180 login: student
```

```
Password:
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$ pwd
```

```
/home/student
```

```
student@wftlab-180:~$ ls
```

System operacyjny GNU/Linux: CLI

```
Debian GNU/Linux wftlab-180 tty1
```

```
wftlab-180 login: student
```

```
Password:
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$ pwd
```

```
/home/student
```

```
student@wftlab-180:~$ ls
```

```
dokumenty gry na zalke test123 zadanie1
```

```
student@wftlab-180:~$
```

System operacyjny GNU/Linux: CLI

```
Debian GNU/Linux wftlab-180 tty1
```

```
wftlab-180 login: student
```

```
Password:
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$
```

```
student@wftlab-180:~$ pwd
```

```
/home/student
```

```
student@wftlab-180:~$ ls
```

```
dokumenty gry na zalke test123 zadanie1
```

```
student@wftlab-180:~$
```

Uruchamianie programu – składnia:

program [OPCJA] [ARGUMENT]

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$ ls
```


System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$ ls zadanie1
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$ ls zadanie1  
main.c notatki.ascii  
student@wftlab-180:~$
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$ ls zadanie1
```

```
main.c notatki.ascii
```

```
student@wftlab-180:~$ ls -l zadanie1
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$ ls zadanie1
main.c notatki.ascii
student@wftlab-180:~$ ls -l zadanie1
razem 0
-rw-r--r-- 1 student student 0 wrz 1 23:20 main.c
-rw-r--r-- 1 student student 0 wrz 1 23:20 notatki.ascii
student@wftlab-180:~$
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$ ls zadanie1
```

```
main.c notatki.ascii
```

```
student@wftlab-180:~$ ls -l zadanie1
```

```
razem 0
```

```
-rw-r--r-- 1 student student 0 wrz 1 23:20 main.c
```

```
-rw-r--r-- 1 student student 0 wrz 1 23:20 notatki.ascii
```

```
student@wftlab-180:~$ ls -a zadanie1
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$ ls zadanie1
main.c notatki.ascii
student@wftlab-180:~$ ls -l zadanie1
razem 0
-rw-r--r-- 1 student student 0 wrz 1 23:20 main.c
-rw-r--r-- 1 student student 0 wrz 1 23:20 notatki.ascii
student@wftlab-180:~$ ls -a zadanie1
. . .x main.c notatki.ascii
student@wftlab-180:~$
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$ ls zadanie1
main.c notatki.ascii
student@wftlab-180:~$ ls -l zadanie1
razem 0
-rw-r--r-- 1 student student 0 wrz 1 23:20 main.c
-rw-r--r-- 1 student student 0 wrz 1 23:20 notatki.ascii
student@wftlab-180:~$ ls -a zadanie1
. . .x main.c notatki.ascii
student@wftlab-180:~$
```

.x	plik/katalog ukryty
..	katalog nadrzędny
.	katalog bieżący
~	katalog domowy

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$
```


System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$ cd
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$ cd zadanie1
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$ cd zadanie1
```

```
student@wftlab-180:~/zadanie1$
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$ cd zadanie1
```

```
student@wftlab-180:~/zadanie1$ ls -l main.c
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$ cd zadanie1
student@wftlab-180:~/zadanie1$ ls -l main.c
-rw-r--r-- 1 student student 0 wrz 1 23:20 main.c
student@wftlab-180:~/zadanie1$
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$ cd zadanie1
student@wftlab-180:~/zadanie1$ ls -l main.c
-rw-r--r-- 1 student student 0 wrz 1 23:20 main.c
student@wftlab-180:~/zadanie1$
```

Użytkownicy należą do grup. Prawa dostępu mogą dotyczyć indywidualnych użytkowników, całych grup lub wszystkich razem.

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$ cd zadanie1
student@wftlab-180:~/zadanie1$ ls -l main.c
-rw-r--r-- 1 student student 0 wrz 1 23:20 main.c
student@wftlab-180:~/zadanie1$
```

Użytkownicy należą do grup. Prawa dostępu mogą dotyczyć indywidualnych użytkowników, całych grup lub wszystkich razem.

Format praw dostępu – 4 bloki

- rwx rwx rwx

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~$ cd zadanie1
student@wftlab-180:~/zadanie1$ ls -l main.c
-rw-r--r-- 1 student student 0 wrz 1 23:20 main.c
student@wftlab-180:~/zadanie1$
```

Użytkownicy należą do grup. Prawa dostępu mogą dotyczyć indywidualnych użytkowników, całych grup lub wszystkich razem.

Format praw dostępu – 4 bloki

- rwx rwx rwx

- 1. blok (-): znak specjalny (plik/katalog),
- 2. blok (rwx): uprawnienia właściciela pliku,
- 3. blok (rwx): uprawnienia użytkowników w grupie właściciela,
- 4. blok (rwx): uprawnienia wszystkich pozostałych użytkowników.

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~/zadanie1$ ls -l main.c
-rw-r--r-- 1 student student 0 wrz 1 23:20 main.c
student@wftlab-180:~/zadanie1$
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~/zadanie1$ ls -l main.c  
-rw-r--r-- 1 student student 0 wrz 1 23:20 main.c  
student@wftlab-180:~/zadanie1$ chmod
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~/zadanie1$ ls -l main.c
-rw-r--r-- 1 student student 0 wrz 1 23:20 main.c
student@wftlab-180:~/zadanie1$ chmod -r
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~/zadanie1$ ls -l main.c  
-rw-r--r-- 1 student student 0 wrz 1 23:20 main.c  
student@wftlab-180:~/zadanie1$ chmod -r main.c
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~/zadanie1$ ls -l main.c
-rw-r--r-- 1 student student 0 wrz 1 23:20 main.c
student@wftlab-180:~/zadanie1$ chmod -r main.c
student@wftlab-180:~/zadanie1$
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~/zadanie1$ ls -l main.c
-rw-r--r-- 1 student student 0 wrz 1 23:20 main.c
student@wftlab-180:~/zadanie1$ chmod -r main.c
student@wftlab-180:~/zadanie1$
```

Brak komunikatu oznacza: OK – zrobione!

```
student@wftlab-180:~/zadanie1$
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~/zadanie1$ ls -l main.c
-rw-r--r-- 1 student student 0 wrz 1 23:20 main.c
student@wftlab-180:~/zadanie1$ chmod -r main.c
student@wftlab-180:~/zadanie1$
```

Brak komunikatu oznacza: OK – zrobione!

```
student@wftlab-180:~/zadanie1$ ls -l main.c
```

System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~/zadanie1$ ls -l main.c
-rw-r--r-- 1 student student 0 wrz 1 23:20 main.c
student@wftlab-180:~/zadanie1$ chmod -r main.c
student@wftlab-180:~/zadanie1$
```

Brak komunikatu oznacza: OK – zrobione!

```
student@wftlab-180:~/zadanie1$ ls -l main.c
--w----- 1 student student 0 wrz 1 23:20 main.c
student@wftlab-180:~/zadanie1$
```


System operacyjny GNU/Linux: **CLI**

```
student@wftlab-180:~/zadanie1$ ls -l main.c
-rw-r--r-- 1 student student 0 wrz 1 23:20 main.c
student@wftlab-180:~/zadanie1$ chmod -r main.c
student@wftlab-180:~/zadanie1$
```

Brak komunikatu oznacza: OK – zrobione!

```
student@wftlab-180:~/zadanie1$ ls -l main.c
--w----- 1 student student 0 wrz 1 23:20 main.c
student@wftlab-180:~/zadanie1$
```

Uruchamianie skryptów:

tworząc nowy skrypt będziemy najczęściej nadawać mu prawo wykonywalności, choć nie jest to konieczne – szczegóły na zajęciach w pracowni.

Wkrótce w pracowni. . .

- i Podstawy obsługi systemu (terminal, narzędzia, instrukcje *bash*);
- ii poruszanie się po systemie plików;
- iii edytor tekstu ***vim***;
- iv tworzenie/uruchamianie skryptów powłoki *bash*;
- v ćwiczenia z nabytych umiejętności.

Materiały do zajęć w pracowni komputerowej:

lista wykonywanych ćwiczeń wraz z wyszczególnieniem używanych narzędzi będzie udostępniana po danych zajęciach. Podane będą też źródła dokumentacji *on-line*.

Pytania i dyskusja