



# Linux

Trener: Maciej Krajewski

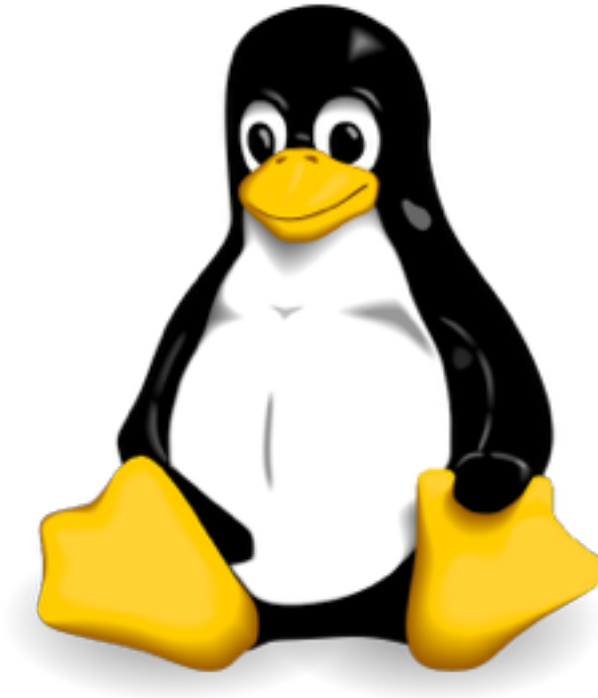
Gdańsk, 1-2 lipca 2016 roku

[www.infoshareacademy.com](http://www.infoshareacademy.com)

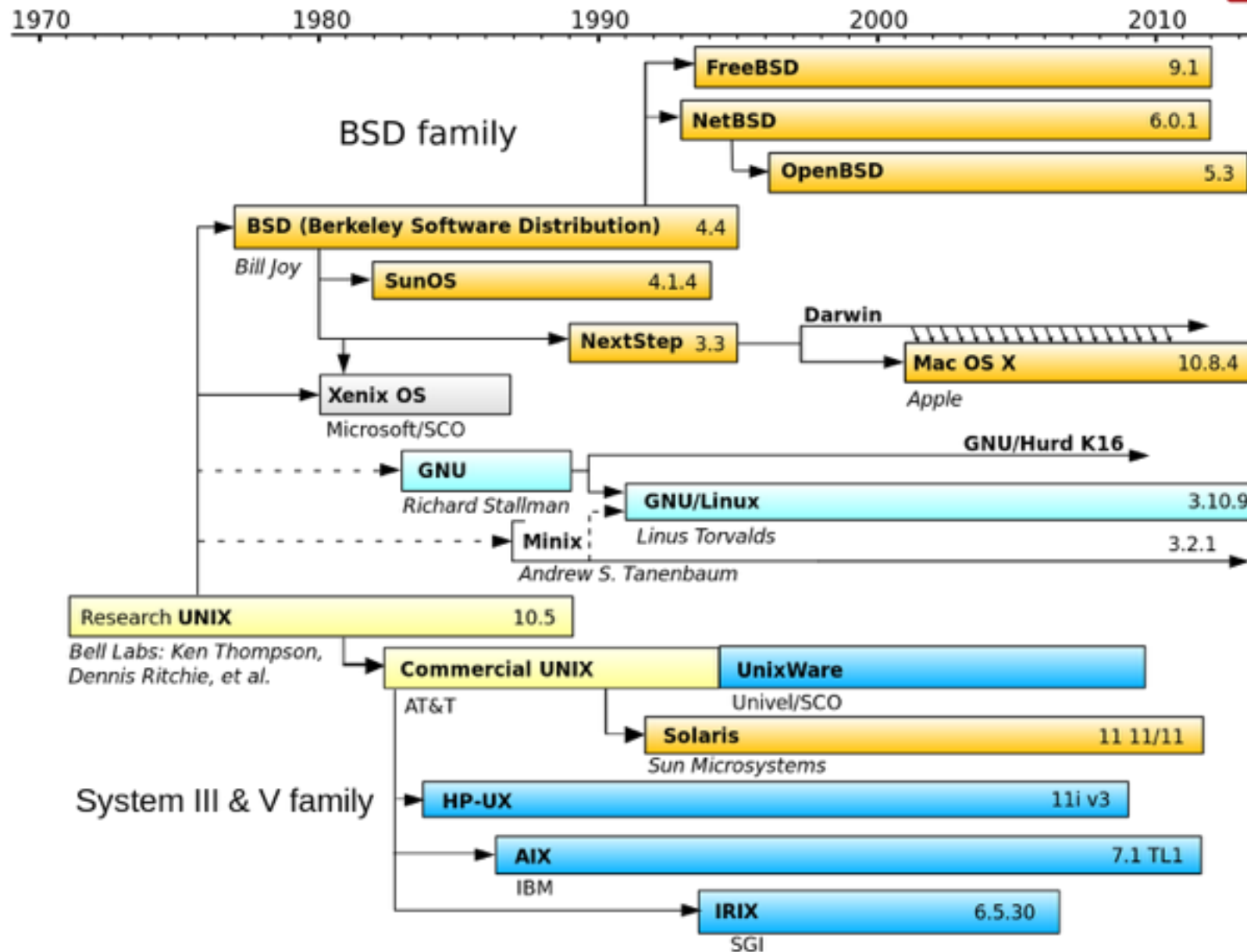
# Agenda

- File System Structure
- zarządzanie użytkownikami
- grupy i uprawnienia
- dowiązania
- instalacja i zarządzanie pakietami
- zarządzanie aplikacjami
- narzędzia
- tunelowanie
- sudo

# Linux



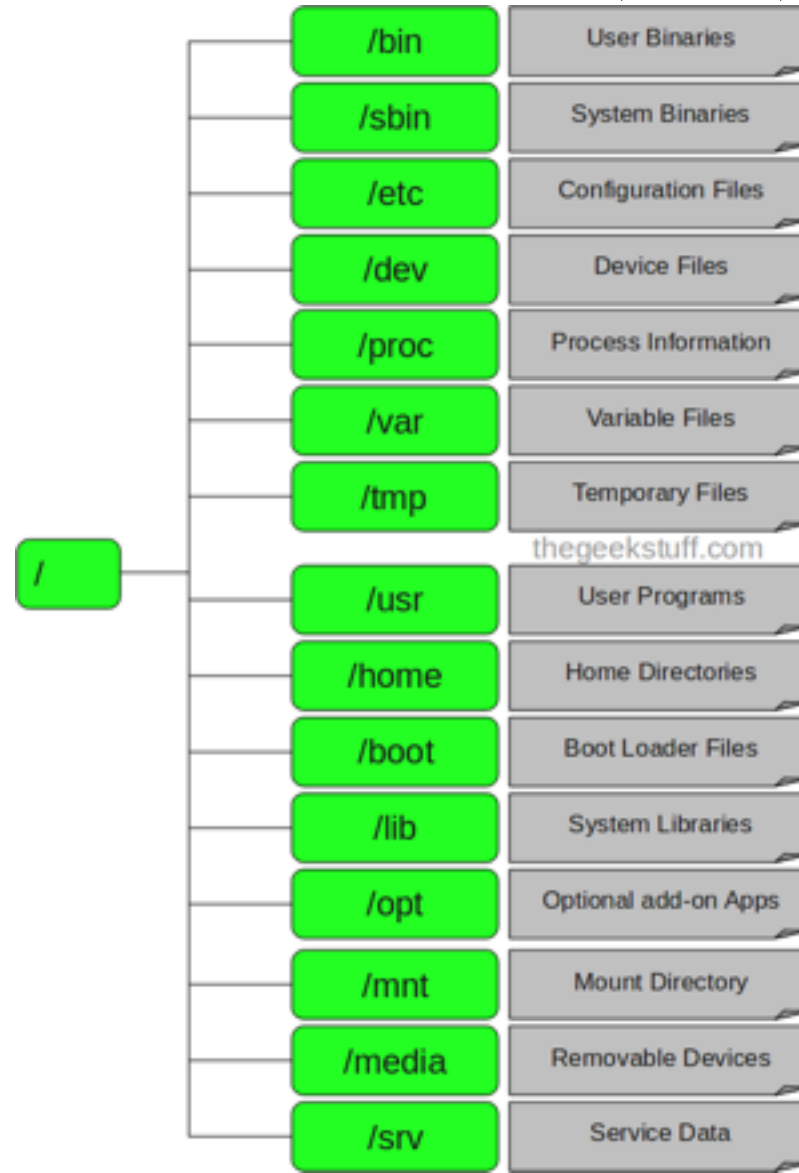
# Linux



# Linux - uruchomienie środowiska

<https://github.com/infoshareacademy/jfdz2-materialy/blob/master/Linux/README.md>

# Linux - File System Structure (FSS)



# Linux - pomoc w poleceniach

*man komenda\**

*komenda -h*

*komenda --help*



# Linux - zarządzanie prawami użytkownika

TYP ZAPISU	Prawo odczytu	Prawo Zapisu	Prawo wykonywania
Potęga 2	$2^2$	$2^1$	$2^0$
Wartość w dziesiętkowym systemie liczbowym	4	2	1
Znak	r (read)	w (write)	x (execute)



# Linux - zarządzanie prawami użytkownika

u – użytkownik (ang. user)

g – grupa (ang. group)

o – inni (ang. others)

a – wszyscy (ang. all)

+ jeżeli chcemy dodać uprawnienia

– jeżeli chcemy odebrać uprawnienia

= jeżeli chcemy zmienić uprawnienia

chmod a+w plik.txt – nadaje wszystkim uprawnienia do zmiany 'plik.txt',  
chmod o-x plik.txt – usuwa możliwość wykonywania 'plik.txt' przez pozostałych użytkowników,  
chmod go=rx plik.txt – grupa oraz pozostali użytkownicy będą mogli tylko czytać i wykonywać.

# §

## Ćwiczenie #1

**Stworzyć 3 użytkowników i korespondujące im katalogi domowe: Adam, Magda, Wojtek  
Utworzyć grupy: „chłopaki” oraz „dziewczyny”**

**Stworzyć katalogi: /opt/group1 oraz /opt/group2 w obydwóch właścicielem jest root, grupa odpowiednio chłopaki i dziewczyny (z opcją zapisu i odczytu i wykonywania)**

**Użytkownicy Adam i Wojtek mają stworzyć pliki w „group1” Magda w „group2”.**

**Dodać Wojtka do grupy Adam.**

**Wojtek ma stworzyć katalog w katalogu domowym Adama, a w nim plik i dać uprawnienia do katalogu i pliku 700**

**Adam ma usunąć katalog stworzony przez Wojtka.**

# Linux - dowiązania

ln -s LOKALIZACJA POWIĄZANIE - Soft link

ln LOKALIZACJA POWIĄZANIE - Hard link

# Linux - zarządzanie pakietami

# Linux - zarządzanie aplikacjami

# Linux - przydatne narzędzia

# Linux - podstawowe wyrażenia regularne

```
^ - początek linii  
$ - koniec linii  
[a-z] - litera od a do z  
[0-9] - cyfra od 0 - 9  
. - dowolny znak  
- nieskończona liczba powtórzeń ostatniego znaku  
\ - „wyescapowany znak” - traktuj kolejny znak dosłownie  
{x} - x powtórzeń poprzedniego ciągu
```

<https://regex101.com>

# §

## Ćwiczenie #2

**Sklonować repozytorium „materialy-admin” z github’a na kontenerze.**

**Rozpakować plik „Linux/logs/apache.log.gz”**

**Przejrzeć plik i powiedzieć ile było unikalnych źródłowych adresów IP**



# Linux - przydatne narzędzia

## Ćwiczenie #3

**Stworzyć wpis w cron który będzie wykonywał polecenie `touch /root/,,`date`` jako user „root” tylko o pełnych godzinach, we wtorek i czwartek między godzinami 9 a 17**

# Linux - przydatne narzędzia

## Ćwiczenie #4

**Stworzyć wpis w cron który będzie wykonywał polecenie „df -h” jako user „root” co minutę od poniedziałku do piątku w godzinach 7:30 do 16:30.**

# Linux - tunelowanie

Local port forwarding

```
ssh -L PORT_LOKALNY:IP_ZDALNE:PORT_ZDALNY USER@IP
```

Remote port forwarding

```
ssh -R PORT_LOKALNY:IP_LOKALNE:PORT_ZDALNY USER@IP
```

# Linux - sudo

```
# User alias specification
User_Alias AN = andrzej

# Cmnd alias specification
Cmnd_Alias AP = /usr/bin/service apache2 *

# User privilege specification
AN ALL = NOPASSWD : AP
```



# Linux - bash

<https://github.com/infoshareacademy/jfdz2-materialy/blob/master/Linux/bash.md>

# Linux - bash

```
#!/bin/bash  
#Tu jest komentarz.  
echo "Hello world"
```

```
chmod +x skrypt.sh  
./skrypt.sh
```

# Linux - bash - echo

```
echo "jakiś napis" > plik  
echo "jakiś napis" >> plik
```



# Linux - bash - zmienne

```
#!/bin/bash  
x=2  
echo "Wartość zmiennej x to $x"
```

# Linux - bash - cytowanie

W bash'u wyróżniamy 3 różne cytowania:

""" - bez zmian

` ` - polecenie

" - „dosłownie”

# Linux - bash - specjalne zmienne

\$0 - nazwa bieżącego skryptu

\$1....9 - nazwa 1....9 zmiennej przekazanej do skryptu

\$@ - wszystkie zmienne przekazane do skryptu

\$? - kod wyjścia ostatnio wykonanego polecenia

# Linux - bash

## Ćwiczenie #5

**Napisać skrypt „loguj.sh” który:**

- **będzie zawierał 2 parametry - imię oraz nazwę pliku z logiem**
- **wypisze przekazane imię na ekran i zapisze je do pliku /var/log/(nazwa pliku z logiem)**

# Linux - bash - if

```
#!/bin/bash
if [ -e ~/.bashrc ]
then
    echo "Masz plik .bashrc"
fi
```

- **-a** plik istnieje
- **-b** plik istnieje i jest blokowym plikiem specjalnym
- **-c** plik istnieje i jest plikiem znakowym
- **-e** plik istnieje
- **-h** plik istnieje i jest linkiem symbolicznym
- **=** sprawdza czy wyrażenia są równe
- **!=** sprawdza czy wyrażenia są różne

- **-n** wyrażenie ma długość większą niż 0
- **-d** wyrażenie istnieje i jest katalogiem
- **-z** wyrażenie ma zerową długość
- **-r** można czytać plik
- **-w** można zapisywać do pliku
- **-x** można plik wykonać
- **-f** plik istnieje i jest plikiem zwykłym
- **-p** plik jest łączem nazwanym
- **-N** plik istnieje i był zmieniany od czasu jego ostatniego odczytu
- **plik1 -nt plik2** plik1 jest nowszy od pliku2
- **plik1 -ot plik2** plik1 jest starszy od pliku2
- **-lt** mniejsze niż
- **-gt** większe niż
- **-ge** większe lub równe
- **-le** mniejsze lub równe

# Linux - bash - case

```
#!/bin/bash
echo "Podaj cyfrę dnia tygodnia"
read d
case "$d" in
    "1") echo "Poniedziałek" ;;
    "2") echo "Wtorek" ;;
    "3") echo "Środa" ;;
    "4") echo "Czwartek" ;;
    "5") echo "Piątek" ;;
    "6") echo "Sobota" ;;
    "7") echo "Niedziela" ;;
    *) echo "Nic nie wybrałeś"
esac
```

# Linux - bash - for

```
for x in jeden dwa trzy  
do  
    echo "To jest $x"  
done
```

# Linux - bash

## Ćwiczenie #6

**Napisać skrypt „find.sh” który:**

- **będzie zawierał 1 parametr**
- **wypisze na ekranie wszystkie pliki znajdujące się w katalogu który jest przekazany jako parametr**



# Linux - bash - while i until

```
#!/bin/bash
x=1;
while [ $x -le 10 ]; do
echo "Napis pojawił się po raz: $x"
x=$((x + 1))
done
```

```
#!/bin/bash
x=1;
until [ $x -ge 10 ]; do
echo "Napis pojawił się po raz: $x"
x=$((x + 1))
done
```

Dziękuję za uwagę