



AKADEMIA GÓRNICZO-HUTNICZA  
IM. STANISŁAWA STASZICA W KRAKOWIE

# Systemy operacyjne

*Unix/Linux*

**dr inż. Andrzej Opaliński**

**pawilon B5/p.413**

**tel. (+48)12 617 53 23**

**e-mail: [andrzej.opalinski@agh.edu.pl](mailto:andrzej.opalinski@agh.edu.pl)**

**Wydział Inżynierii Metali i Informatyki Przemysłowej**

Nazwa	Twórca	Data pierwszego wydania	Poprzednik	Wersja stabilna	Cena w USD	Preferowana licencja	Przeznaczenie
Arch Linux	Judd Vinet	11 marca 2002	CRUX	2008.06	Darmowy	GPL	stacje robocze, użytkownicy domowi
Aurox	Robert Głowczyński	listopad 2002	Red Hat Linux	12.0	Darmowy	GPL	stacje robocze, użytkownicy domowi
Debian	Debian Project	sierpień 1993	SLS	5.0 (lenny)	Darmowy	każda zgodna z WDdWO	serwery, stacje robocze, użytkownicy domowi
Fedora	Fedora	5 listopada 2003	Red Hat Linux	11	Darmowy	GPL	serwery, stacje robocze, użytkownicy domowi
Knoppix	Klaus Knopper	19 stycznia 2003	Debian	5.3.1	Darmowy	GPL	stacje robocze, użytkownicy domowi
Ubuntu	Canonical Ltd	październik 2004	bazujący na Debianie	9.04	Darmowy	GPL	serwery, stacje robocze, użytkownicy domowi
Gentoo	Gentoo Foundation, Inc	marzec 2002	Enoch	2008.0	Darmowy	GPL	serwery, stacje robocze, użytkownicy domowi
Linspire	Linspire Inc.	marzec 2002	Lindows	6.0	Płatny	GPL, Free software, Oprogramowanie własnościowe	stacje robocze, użytkownicy domowi
Mandriva Linux	Mandriva	lipiec 1998	Red Hat Linux	2008.1	Płatny / darmowy - okrojony	GPL	stacje robocze, użytkownicy domowi
MEPIS	Warren Woodford	wrzesień 2003	bazuje na Debianie	7.0	Darmowy	GPL	serwery, stacje robocze, użytkownicy domowi
Slackware Linux	Patrick Volkerding	lipiec 1993	SLS	12.1	Darmowy	GPL	serwery, stacje robocze, użytkownicy domowi
openSUSE	Novell	marzec 1994	Jurix	11.1	Darmowy	GPL	serwery, stacje robocze, użytkownicy domowi

# LOGO



**Debian** - projekt wolnej dystrybucji systemu operacyjnego GNU/Linux realizowany przez ochotników na całym świecie.

**Debian** cieszy się opinią stabilnego systemu o wysokiej jakości i łatwego do aktualizacji. Ze względu na dbałość o jakość i bezpieczeństwo dystrybucji, nowe wersje stabilne pojawiają się relatywnie rzadko, często dochodzi też do opóźnień w ich wydawaniu.



**Fedora** – nazwa następcy wolnej dystrybucji Red Hat Linux rozwijanej przez Fedora Project i finansowanej głównie przez Red Hat.

Twórcy Fedory stawiają na innowacyjność, dlatego też kolejne wydania pojawiają się często i zawierają najnowsze dostępne oprogramowanie, nawet jeśli prace nad stabilną wersją nie zostały jeszcze ukończone.

Z tego powodu, oraz faktu ścisłego powiązania z Red Hatem często, lecz niesłusznie Fedorę określa się mianem "poligonu Red Hata".

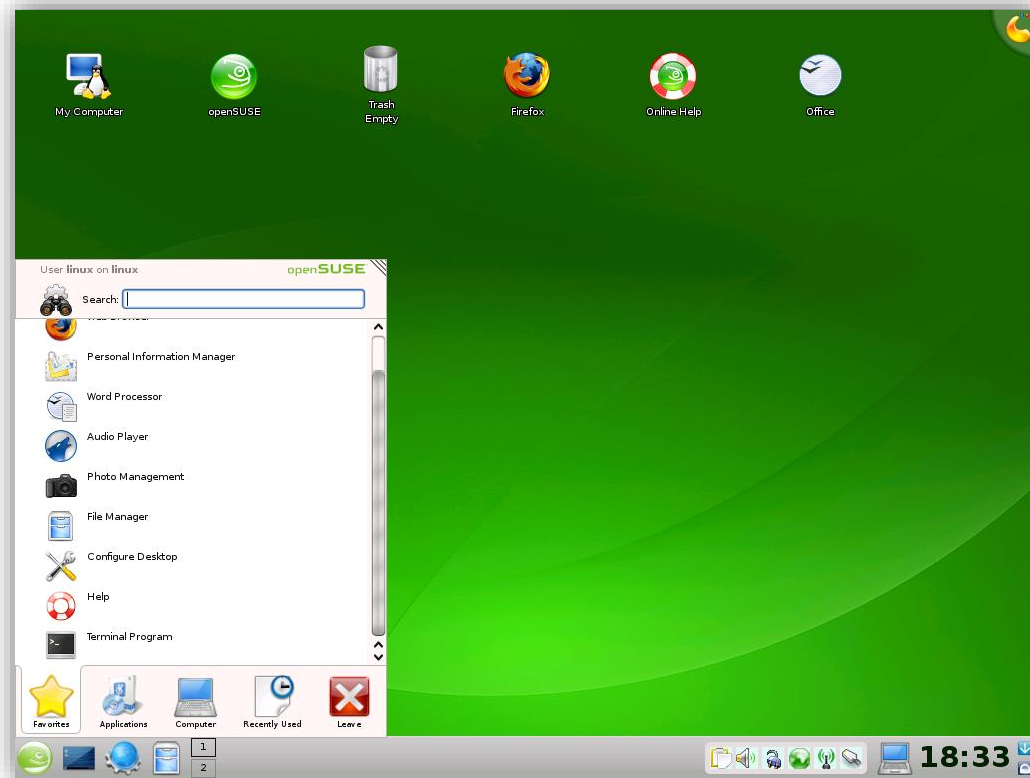
Fedora jest stosowana zarówno jako system operacyjny dla komputerów domowych, jak i serwerów.





**SUSE Linux** – dystrybucja Linuksa, stworzona z myślą o użytkownikach korzystających na co dzień z interfejsu graficznego i przeznaczona zarówno dla użytkowników domowych (openSUSE), jak i do zastosowań komercyjnych (SUSE Linux Enterprise).

Rozwijana początkowo w Niemczech przez SuSE GmbH, w styczniu 2004 przejęta przez firmę Novell.



PLD (na początku Polish(ed) Linux Distribution, obecnie akronim rekurencyjny od ang. PLD Linux Distribution), tworzona w Polsce dystrybucja systemu Linux.

Jest to dystrybucja przeznaczona dla zaawansowanych użytkowników, oparta na pakietach oprogramowania RPM, przystosowana do łatwego i szybkiego uaktualniania za pomocą narzędzia poldek.



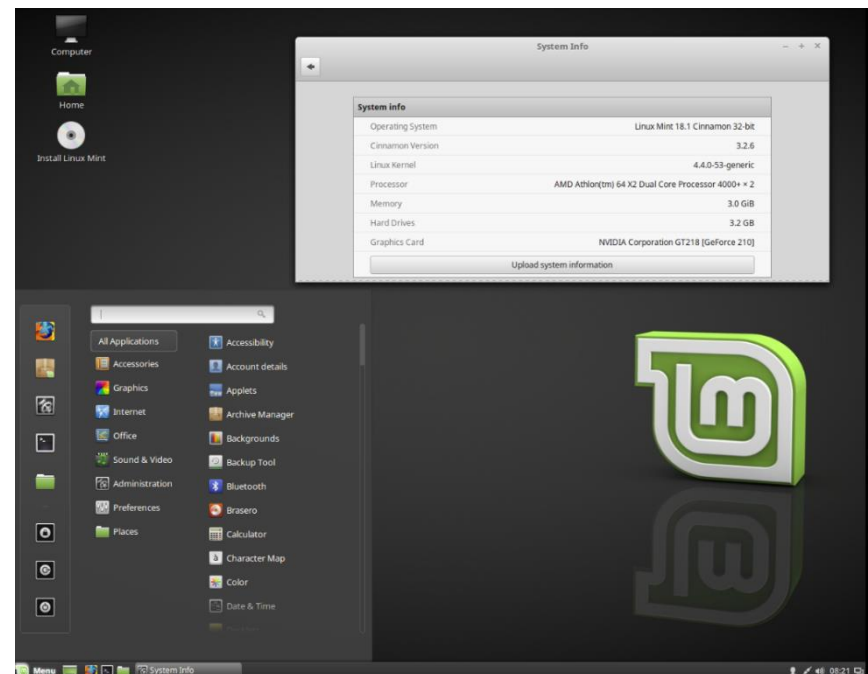


Linux Mint – dystrybucja GNU Linux oparta na Ubuntu i Debianie, skierowana do początkujących użytkowników.

Dystrybucja kładzie nacisk na prostotę użytkowania, przydatne aplikacje i pełne ich wsparcie zaraz po instalacji.

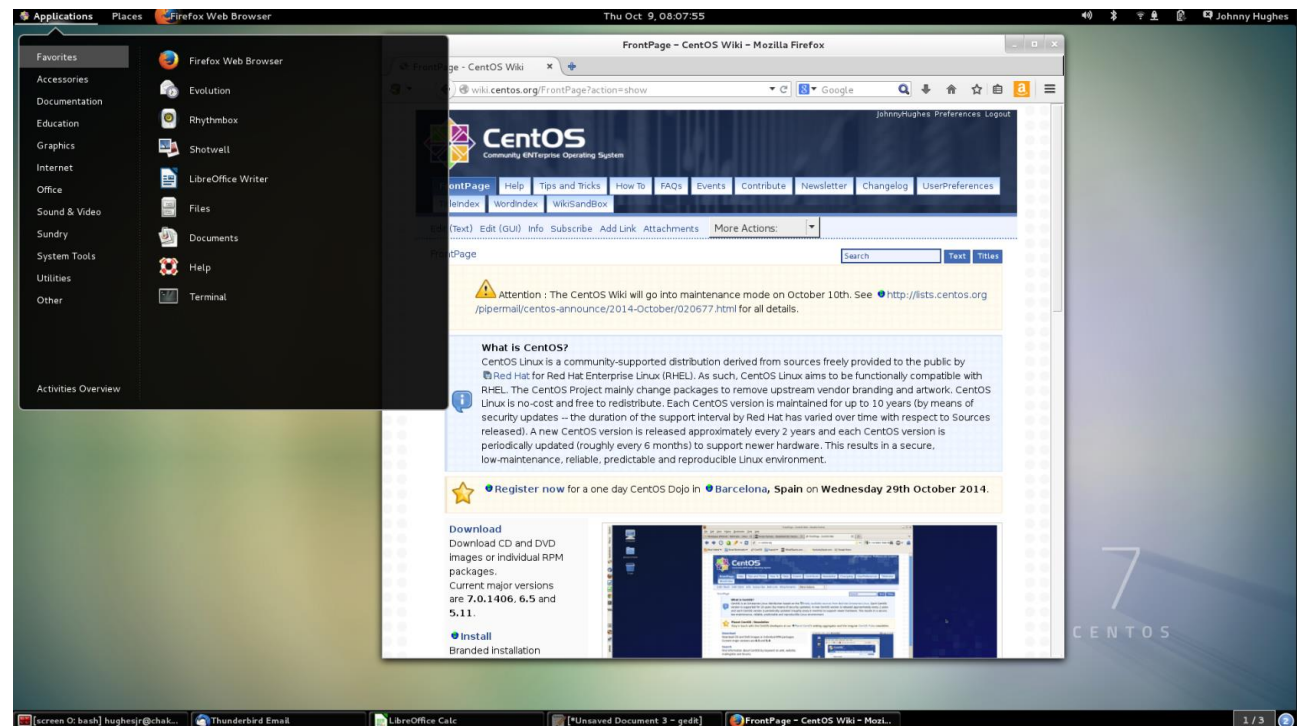
W przeciwieństwie do Ubuntu obsługuje standardowo wiele oprogramowania własnościowego, którego nie trzeba samodzielnie instalować.  
(Flash, Java, kodeki)

Dostępna także jako LiveCD/USB



CentOS – dystrybucja Linuxa oparta na RedHat Enterprise Linux i mająca być w nim całkowicie kompatybilna

Do zarządzania pakietami – yum, up2date



**KDE (K Desktop Environment)** – środowisko graficzne zrealizowane jako wolne oprogramowanie dla systemów operacyjnych Unix i GNU/Linux. Zawiera bogaty system bibliotek do tworzenia aplikacji w środowisku graficznym.

**KDE** korzysta z biblioteki graficznej Qt, a także z systemu obiektowego KParts. System ten pozwala na współdzielenie elementów danego programu pomiędzy różnymi aplikacjami KDE, np. emulator terminala Konsole może zostać osadzony w Konquerorze.

W **KDE** menedżerem okien jest **KWin**, jednak aplikacje można uruchamiać także pod innymi menedżerami.



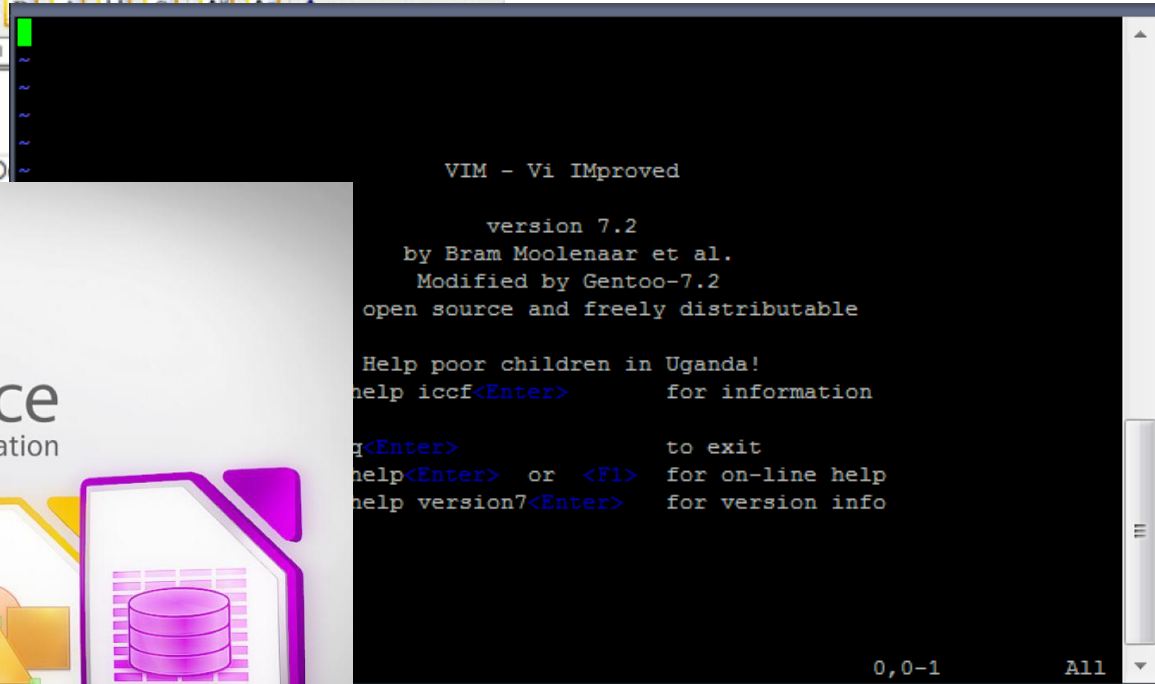
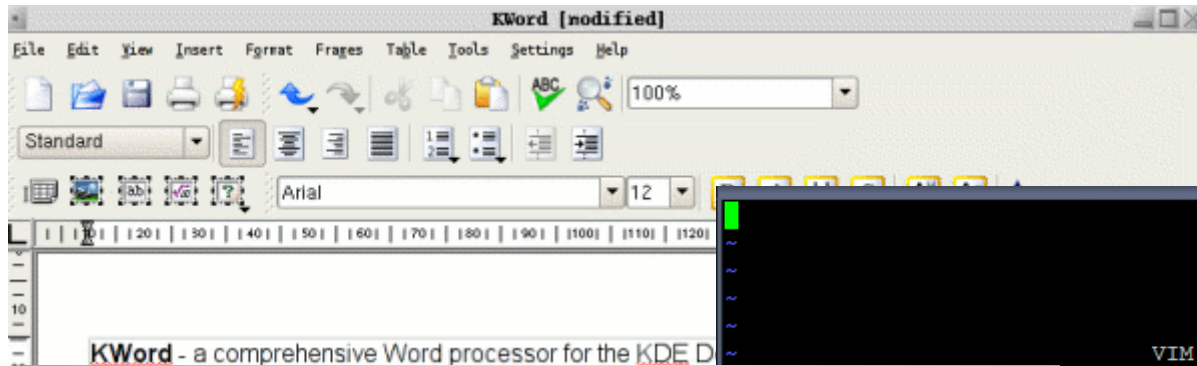
# Środowisko graficzne GNOME

**GNOME** (**GNU Network Object Model Environment**) - graficzne środowisko użytkownika oparte na systemie X Window.

Projekt został rozpoczęty w sierpniu 1997 w celu stworzenia w pełni wolnodostępnego graficznego środowiska użytkownika dla systemu GNU/Linux, **jako alternatywa dla środowiska KDE**, opartego o bibliotekę Qt

Większa część środowiska **GNOME** została napisana w czystym C.





X MULTIMEDIA SYSTEM

00:11 5. RAMMSTEIN - DU HAST (3:55)  
192 kbps 44 kHz

X MULTIMEDIA SYSTEM - EQUALIZER

ON AUTO PRESETS

PREAMP 60 170 310 600 1K 3K 6K 12K 14K 16K

PLAYLIST EDITOR

1. Rammstein - Alter Mann 4:24
2. Rammstein - Angel 4:25
3. Rammstein - Bestrafe Mich
4. Rammstein - Buck Dich
5. Rammstein - Du Hast
6. Rammstein - Eifersucht
7. Rammstein - Engel
8. Rammstein - Klavier
9. Rammstein - Kuss Mich [Fellfrosch]
10. Rammstein - Sehnsucht
11. Rammstein - Spiel Mit Mir
12. Rammstein - Tier
13. Rammstein - You Hate
14. Metallica - And Justice For All
15. Metallica - Battery
16. Metallica - Blackened
17. Metallica - Creeping Death
18. Metallica - Fade To Black
19. Metallica - Fight Fire With Fire
20. Metallica - For Whom The Bell Tolls
21. Metallica - Hit The Lights
22. Metallica - Jump In The Fire
23. Metallica - Master Of Puppets
24. Metallica - One
25. Metallica - Phantom Load
26. Metallica - Ride The Lightning
27. Led Zeppelin - Led Zeppelin-Black dog
28. Led Zeppelin - Led Zeppelin-Rock and...
29. Led Zeppelin - Led Zeppelin-Stairway ...

FILE FILE SEL ALL MISC OPT. 3:55/2:38:17 88:11

KMPlayer Build 1434

Open Album Art  
"Convenient viewing experience and pleasure"

KMPlayer

02:23

CONTROL

PLAYLIST

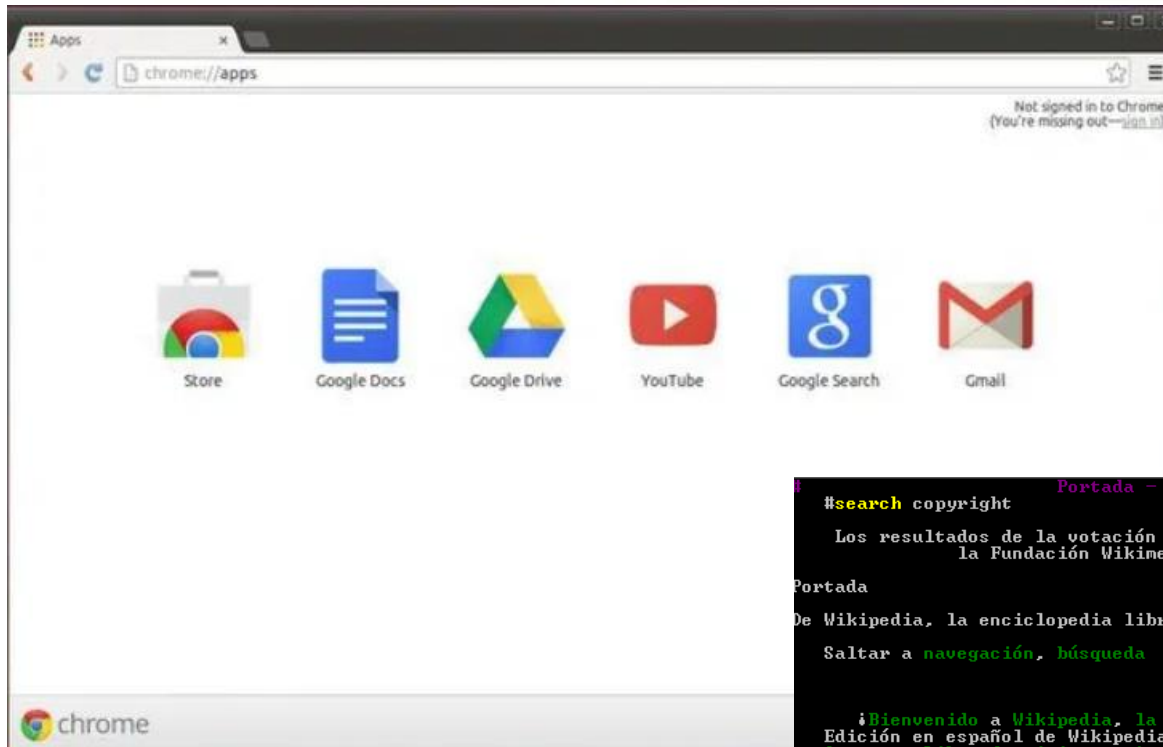
22. Both Sides Now.mp3
23. Boyzone - Everyday I love you.mp3
24. Bread - If (71).mp3
25. Brenda Russell - Le Restaurant.mp3
26. Bryan Adams - Heaven.mp3
27. Carole King - You've Got A Friend.mp3
28. Cat Stevens - Morning Has Broken.mp3
29. Chris De Burgh - A space man came travelling.mp3
30. Chris De Burgh - The Girl With April In Her Eyes.mp3
31. Charlene - Ode to my Family.mp3
32. Our Forebears - Longue.mp3
33. Deep Purple - Soldier of Fortune.mp3
34. Denis Roussel - Stand by me.mp3

MPlayer MPlayer - 21:25

00:00:00 00 - No media opened. [00]

MPlayer - Video

MPlayer



```

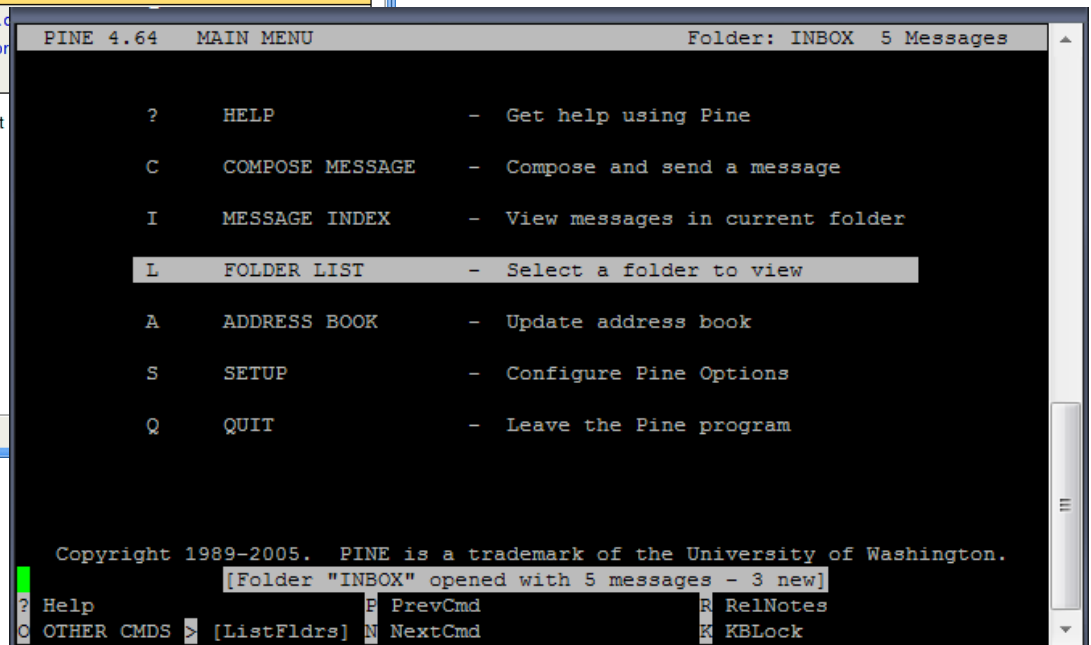
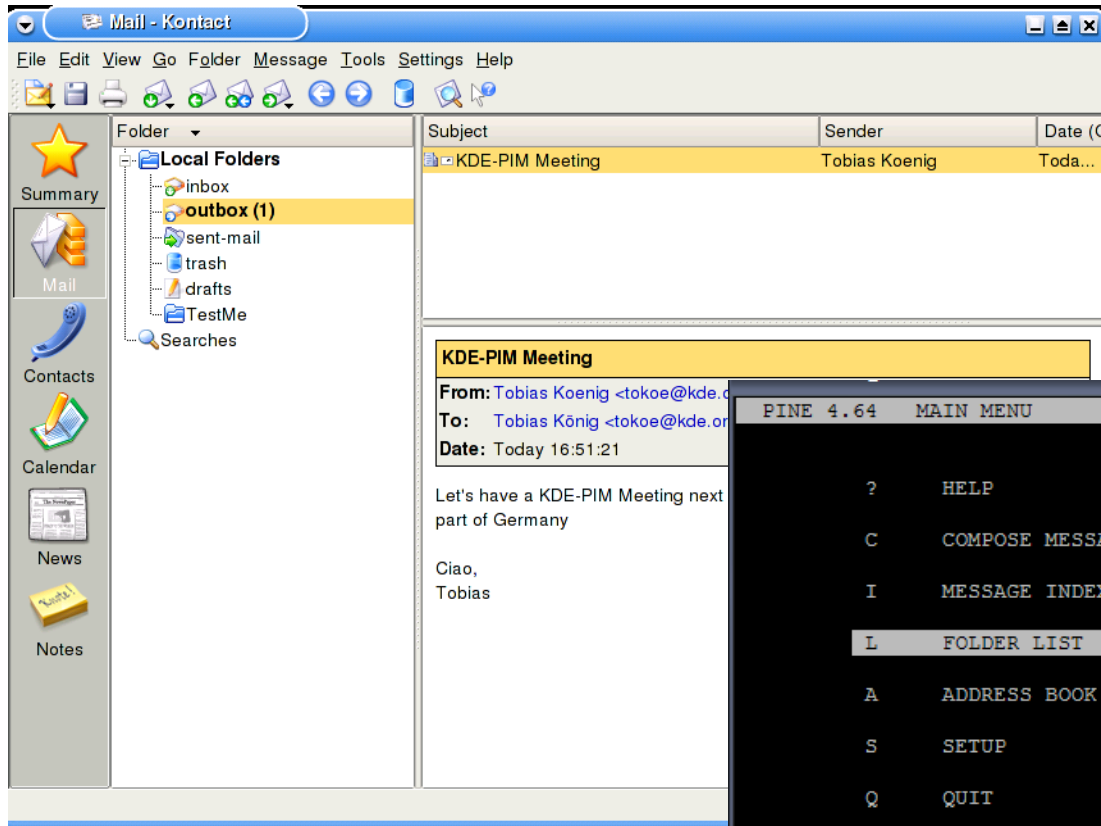
Portada - Wikipedia, la enciclopedia libre <pl of 11>
#search copyright
Los resultados de la votación para el Consejo de Administración de
la Fundación Wikimedia ya están disponibles.
Portada
De Wikipedia, la enciclopedia libre
Saltar a navegación, búsqueda

¡Bienvenido a Wikipedia, la enciclopedia libre!
Edición en español de Wikipedia, iniciada en el 2001.
Quarto - Libro de visitas - Acceso WAP - Contacto

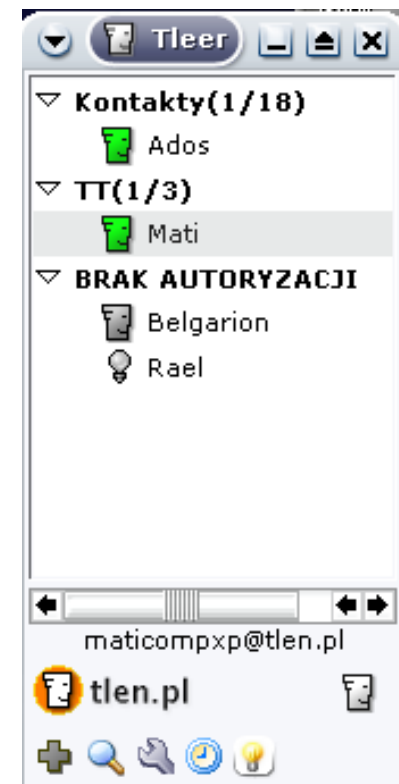
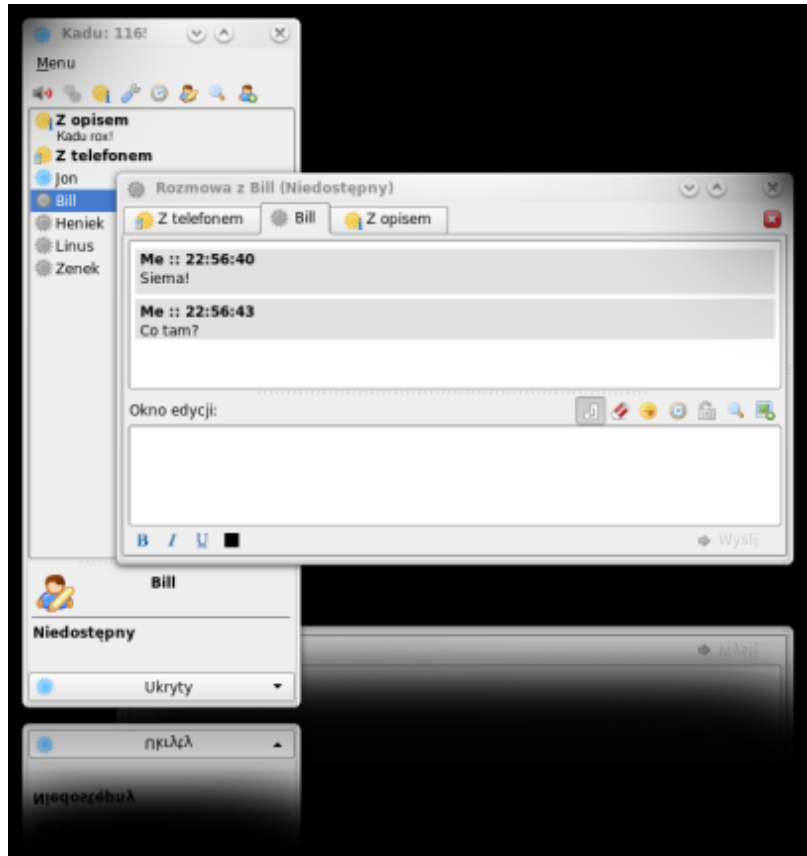
Explora Wikipedia
Exploración Exploración Clasificación Sistemas de clasificación
Búsqueda · Índice alfabético · Índice de categorías · Portales
temáticos · Todos los artículos · Acceso WAP Decimal Universal*
Unesco* <4 dígitos* · 6 dígitos* · Listas <acrónimos · Biografías ·
Países>

Buscar título exacto Buscar en el texto
En la columna de la izquierda de cada artículo hay una caja de
búsqueda equivalente a esta.
(NORMAL LINK) Use right-arrow or <return> to activate.
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list

```

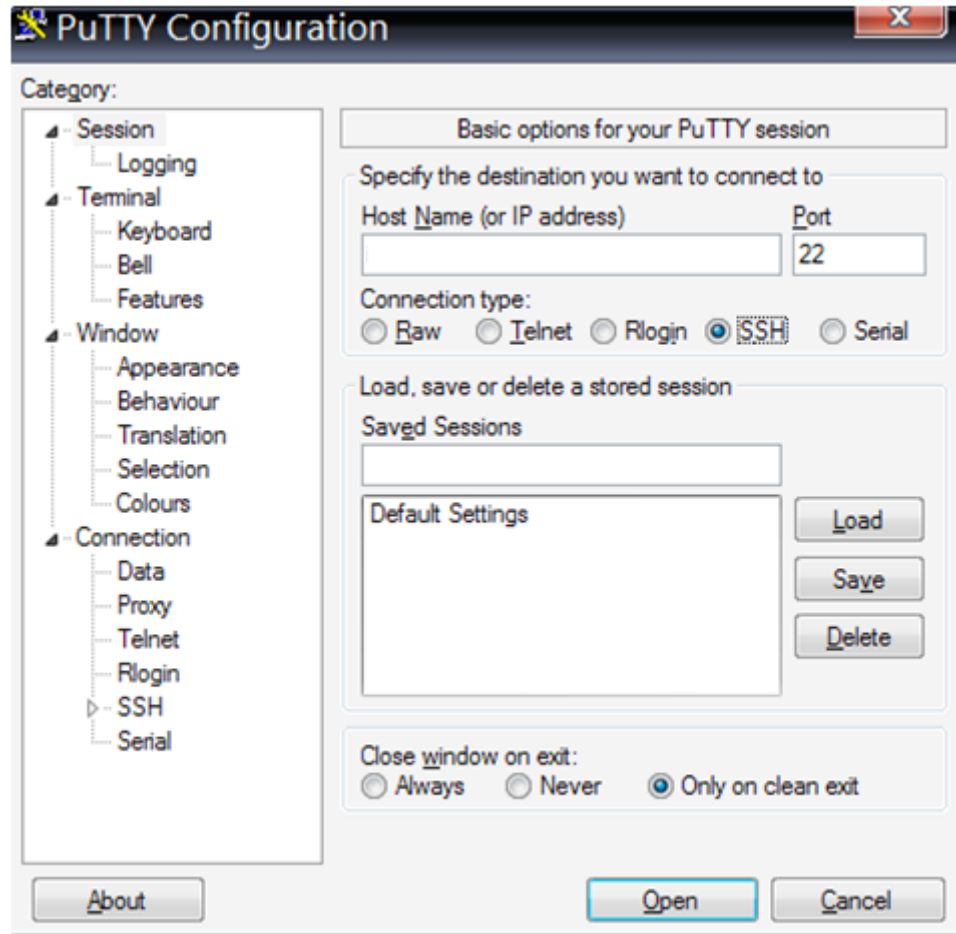






# Nawiązanie połączenia z serwerem

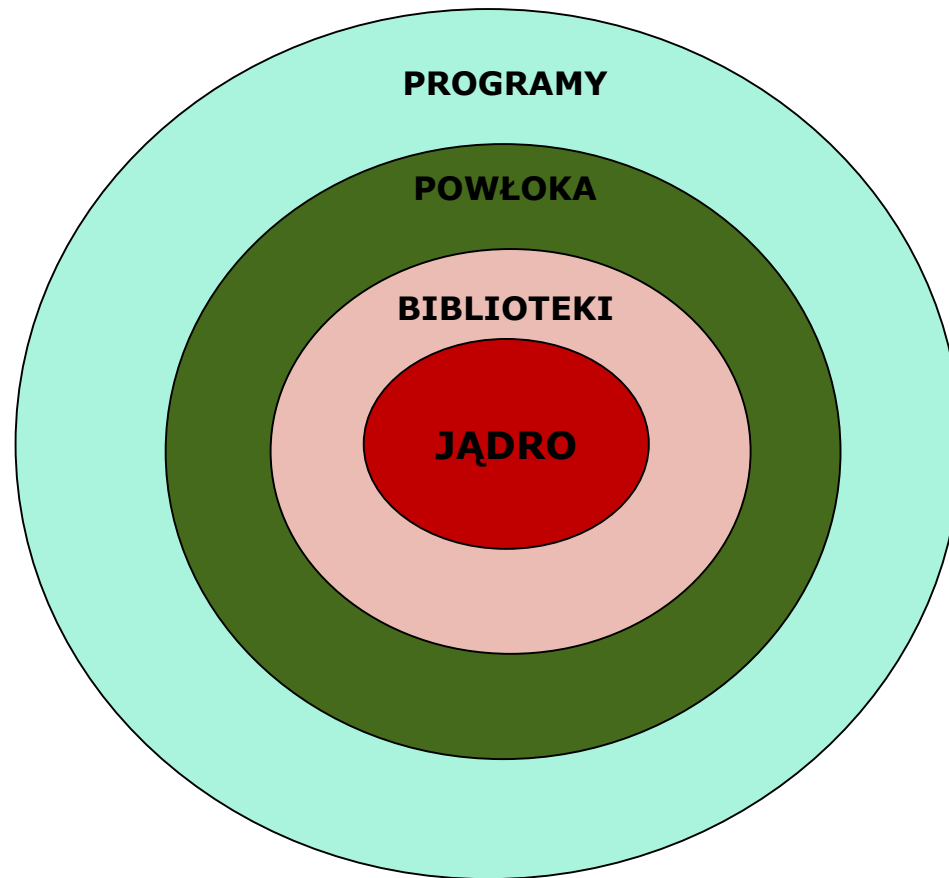
## Program putty



# Linux - model warstwowy systemu

System linux można podzielić **na 4 warstwy**:

- jądro,
- biblioteki,
- powłoka
- programy.



System UNIX/Linux składa się z:

- **jądra,**
- **powłoki**
- **z wielu podsystemów** i programów zapewniających określone usługi np. obsługę systemu plików, urządzeń.

**Jądro** (ang. kernel)

Jądro zawiera zbiór programów - zarządzanie zasobami.

Jądro ma kontrolę nad komputerem, a użytkownik komunikuje się z jądrem przez tzw. **powłokę**.

**Powłoka** (ang. shell)- dostęp do jądra systemu, istnieje wiele powłok (języków powłok) - powłoka Bourne'a (sh), powłoka Korn'a (ksh), powłoka C (csh)

Po zalogowaniu się, system operacyjny umieszcza użytkownika w katalogu osobistym (ang. home directory) i uruchamia program powłoki.

Powłoka przekazuje polecenia użytkownika do jądra.



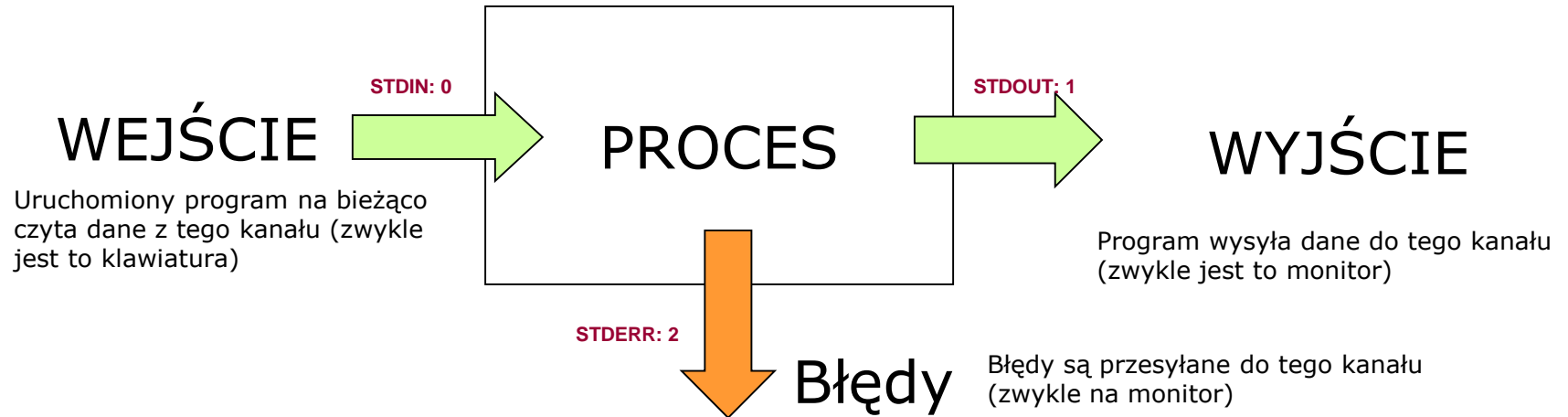
# Linux - model warstwowy systemu

## Powłoka:

- nazwa pochodzi stąd, że warstwa ta oddziela wewnętrzną część systemu operacyjnego od użytkownika,
- powłoka zawiera **interpreter poleceń**, który umożliwia komunikację z użytkownikiem,
- interpreter poleceń **uruchamia polecenia** systemu operacyjnego oraz programy użytkowe.

## Programy:

- procesy uruchamiane przez użytkownika,
- zarządzane przez jądro,
- mogą być przerwane w dowolnym momencie, np. komendą **- kill**
- każdy ma przydzielony odpowiedni obszar pamięci i priorytet,



W systemie Linux procesy mogą komunikować się ze sobą, systemem, bądź użytkownikiem za pomocą trzech kanałów komunikacyjnych:

- Standardowego wejścia (STDIN)
- Standardowego wyjścia (STDOUT)
- Standardowy błąd (STDERR)

Mogą one odwoływać się na przykład do terminala, klawiatury czy pliku. Do każdego z nich możemy odwołać się za pomocą liczb:

- STDIN: 0
- STDOUT: 1
- STDERR: 2

Za pomocą znaków `|`, `<`, `>`, `>>` możemy tworzyć swego rodzaju „kanały/rurociągi” które umożliwiają nam „transfer” danych przez wiele procesów i jednoczesne podawanie danych za ich pośrednictwem „obróbce”.

- `|` – przekazuje **wyjście** jednego procesu na **wejście** drugiego
- `<` – zmienia standardowe **wejście** polecenia (zazwyczaj klawiatura) na **plik**
- `>` – przekierowuje **wyjście do pliku**. Jeśli plik zawiera już jakieś dane to zostają one **nadpisane**
- `>>` – przekierowuje **wyjście do pliku**, przy czym dane wyjściowe są do niego dopisywane, **nie powoduje utraty zawartych** wcześniej w pliku danych
- `2>` – przekierowuje standardowy **kanal błędu** np:  
• ***skrypt1\_zajecia.sh 2> more***
- `>&` – przekierowuje zarówno standardowe **wyjście jak i błąd** w określone miejsce

Wyświetlenie listy powłok:

```
# cat /etc/shells
```

```
/bin/bash
```

```
/bin/csh
```

```
/bin/sh
```

```
/bin/tcsh
```

uruchamiamy np.:

```
csh
```

wychodzimy:

```
exit
```

```
linux@linux:/etc> cat shells  
/bin/ash  
/bin/bash  
/bin/csh  
/bin/dash  
/bin/false  
/bin/ksh  
/bin/ksh93  
/bin/pdksh  
/bin/sh  
/bin/tcsh  
/bin/true  
/bin/zsh  
/usr/bin/csh  
/usr/bin/dash  
/usr/bin/ksh  
/usr/bin/ksh93  
/usr/bin/passwd  
/usr/bin/pdksh  
/usr/bin/bash  
/usr/bin/tcsh  
/usr/bin/zsh  
linux@linux:/etc> █
```



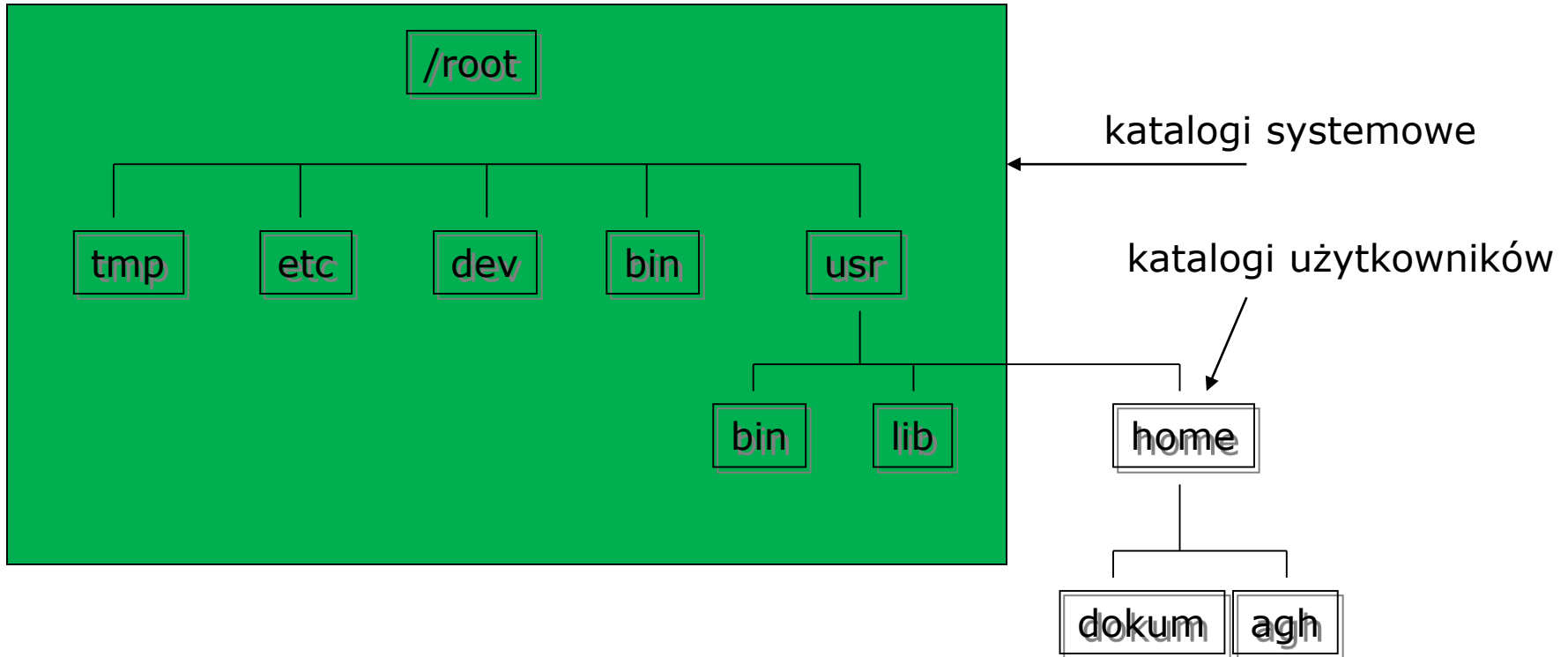
## zmiana powłoki systemowej

- **[marcin@localhost marcin] \$ sh**  
podstawowa powłoka w każdym systemie typu Unix, **sh** jest również językiem skryptowym
- **[marcin@localhost marcin] \$ bash**  
najbardziej popularna powłoka używana na systemach unixowych, jest też skryptowym językiem programowania, umożliwiającym efektywne zarządzanie systemem

Podobny do Windows – jeden katalog główny i drzewiasta, wspólna struktura katalogów.

Jeśli jest kilka dysków fizycznych (także napędów optycznych) nie są one widziane jak w Windows pod literami - mogą być reprezentowane jako osobne katalogi (montowanie - *mount*)

- / symbol katalogu głównego (bez nazwy)
- symbol katalogu bieżącego
- .. symbol katalogu nadrzędnego



Rozbudowana struktura katalogów Linuxa jest pogrupowana w logiczne części. Każdy katalog zawiera pliki powiązane ze sobą logicznie.

## Najważniejsze:

**/bin** - wykonywalne pliki binarne, niezbędne do pracy zarówno w trybie wieloużytkownikowym, jak i w awaryjnym trybie jednego użytkownika,

**/boot** - jądro systemu oraz pliki niezbędne przy jego uruchomieniu,

**/etc** – pliki konfiguracyjne systemu,

**/home** - katalogi domowe użytkowników,

**/lib** - biblioteki systemowe,

**/lost+found** - pliki odnalezione podczas wykonywania testów dysku,

**/mnt** - katalog do którego zwykle montowane są tymczasowe systemy plików

**/root** - katalog domowy użytkownika root,

**/tmp** - pliki tymczasowe,

**/usr** - pliki użytkowników, dodatki doinstalowywane do systemu, id...

**/var** - pliki często zmieniane, np. logi.

**/usr/doc** - dodatkowa dokumentacja,

**/usr/info** - dokumentacja dla polecenia **info**,

**/usr/lib** - biblioteki systemowe,

**/usr/man** - pliki pomocy kontekstowej **man**,

**/usr/sbin** - wykonywalne polecenia administracyjne systemu Linux potrzebne do pracy w trybie wieloużytkownikowym,

**/usr/src** - źródła do systemu i jądra,

**/var/db** - bazy danych,

**/var/lock** - semafore używane do komunikacji międzyprocesowej,

**/var/log** - logi systemowe,

**/var/named** - pliki serwera nazw,

**/var/run** - pliki z danymi dotyczącymi numerów poszczególnych procesów w systemie,

**/var/spool** - kolejki poczty, wydruków itp.

- ext4 - następca ext3, obecnie jeden z najpopularniejszych systemów plików dla Linuksa
- ext2 - dobry na małe, stare dyski
- ext3 - nowocześniejszy, umie obsłużyć nowe, duże dyski, duże partycje i duże pliki
- ReiserFS - zaleta: umie szybko odczytywać duże ilości małych plików, nadaje się na serwer
- ftpfs, smbfs, sshfs - system plików do zarządzania odległymi zasobami (w sieci), udostępnianymi poprzez serwery: FTP, SAMBA, SSH



# Nazwy plików i katalogów

- **rozdzielalne małe i duże litery!** (case sensitive) - czyli mogą być dwa pliki o nazwach x i X (w Windows nie),
- Windows i Linux nie pozwolą na plik i katalog o tej samej nazwie)
- mogą się zaczynać od cyfry, niektóre znaki różne od cyfr i liter są dozwolone, np. \_, kropka (może być wiele kropek w nazwie)
- nazwa zaczynająca się od kropki to plik ukryty
- nie wolno używać SPACJI! wewnątrz nazwy

- przydzielenie przestrzeni dyskowej
- wybór oprogramowania i kopiowanie na dysk
- ustalenie hasła użytkownika *root*
- konfiguracja startu systemu

Opcjonalnie mogą być wykonane następujące czynności:

- lokalizacja systemu (wybór strefy czasowej)
- konfiguracja interfejsu graficznego X Windows
- konfiguracja usług uruchamianych przy starcie systemu (np. ftp, www)

## Logowanie w trybie tekstowym

root - konto administratora

# - znak zachęty

\$ - znak zachęty

**[użytkownik@host katalog]\$ █**

**np.**

**[root@localhost root]# █**

**root** to tradycyjna nazwa uniksowego konta, które ma pełną kontrolę nad systemem.

**Z założenia konto root nie powinno być używane do pracy, do której wystarczyłoby zwykłe konto z ograniczonymi uprawnieniami. Istotną sprawą jest zabezpieczenie tego konta silnym hasłem i zabezpieczenie przed nieautoryzowanym dostępem.**

Dobrze jest ograniczyć możliwość logowania na konto root i używać polece **su** albo **sudo**.

Konto **root** uprawnia do wykonywania takich operacji jak zmiana właściciela pliku czy otwarcie portu TCP/UDP z numerem poniżej 1024. W innych systemach operacyjnych używa się też nazw takich jak **toor, superuser, supervisor, Administrator, czy operator**. Nazwa **root** funkcjonuje, jako określenie administratora systemu, zarówno w systemach UNIX, jak i pokrewnych (FreeBSD, GNU/Linux).

## UID (ang. User Identifier)

- jest to identyfikator użytkownika w systemie Unix.

### Reguły:

- root ma UID 0,
- użytkownik *nobody* ma ostatni UID (zazwyczaj 32767),
- UID-y od 1 do 100 są zarezerwowane dla systemu.

## Atrybuty użytkownika

System przechowuje w pliku **/etc/passwd** następujące atrybuty każdego zarejestrowanego użytkownika:

**nazwa** - nazwa jednoznacznie identyfikująca konto użytkownika,

**identyfikator użytkownika UID** - numer jednoznacznie identyfikujący użytkownika w systemie,

**identyfikator grupy GID** - numer grupy, do której należy użytkownik,

**katalog domowy** - prywatny katalog użytkownika, w którym może bezpiecznie przechowywać swoje pliki, zabezpieczone przed dostępem innych użytkowników,

**powłoka logowania** - nazwa interpretera poleceń, który jest uruchamiany po zalogowaniu użytkownika.

## Logowanie i wylogowanie

- logowanie  
login: marcin  
password: \*\*\*\*\*)
- wylogowanie  
[marcin @localhost marcin] \$ logout
- bezpieczne wyłączenie  
[marcin l@localhost marcin] \$ halt  
[marcin @localhost marcin] \$ shutdown -h now
- restart  
[marcin @localhost marcin] \$ shutdown -r now

## Zmiana hasła

- zmiana hasła użytkownika  
`[marcin@localhost marcin] $ passwd`
- zmiana hasła innym użytkownikom (tylko na koncie roota)  
`[root@localhost root] # passwd marcin`



# **Polecenia podstawowe**

Ogólna składnia polecenia:

**polecenie -opcje parametry**



opcjonalne

`pwd` wyświetla nazwę katalogu bieżącego

Spis pozycji danego katalogu

`ls -l`

wyświetla pełną zawartość katalogu bieżącego – opcja `-l` pełne informacje o prawach, rozmiarze, dacie utworzenia, właścicielu

`ll` – krótki odpowiednik `ls -l` (nie we wszystkich systemach unixowych)

# Poruszanie się po strukturze katalogów

- `cd` ścieżka - zmiana katalogu bieżącego
- `cd` .. - zmiana katalogu bieżącego na nadrzędny
- `cd` / - zmiana katalogu na główny
- `cd` - zmiana katalogu bieżącego na domowy użytkownika (powrót do domowego) lub `cd` ~
- `cd` ../KAT - zmiana katalogu na inną gałąź („wyjdź wyżej i „wejdź” do KAT)

**mkdir** nazwa

- tworzenie katalogu

**rmdir** nazwa

- usunięcie katalogu (pustego!)

Usunięcie katalogu niepustego

**rm -r katalog\_podrz**

Zmiana nazwy katalogu

**mv** nazwa\_stara nazwa nowa

Tworzenie nowego pliku pustego  
`touch plik`

Tworzenie nowego pliku z treścią  
`cat> plik`

*Zawartość pliku...*

.....

*kończymy* `CTRL+D`

**rm nazwa**

- usunięcie pliku

**rm wzorzec**

- usunięcie plików według wzorca

*znaki specjalne wzorca*

\* zastępuje dowolny ciąg znaków

? zastępuje jeden znak

**rm \***

- usuwa wszystkie pliki w bieżącym katalogu

**ls -l p\***

- wyświetla pliki - nazwa na literę p (reszta dowolna)

- `cp plik1 plik2` - źródło i cel w tym samym katalogu (bieżącym)
- `cp ../plik1 plik2` - źródło w katalogu nadrzędnym a cel w tym samym katalogu (bieżącym)
- `cp plik1 ./KAT/plik2` - źródło w katalogu bieżącym a cel w podrzędnym do bieżącego katalogu KAT
- `cp -r KAT ścieżka/KAT2` - kopiowanie rekursywne (katalogu z zawartością)



# Zmiana nazwy (przeniesienie) pliku

unikalna nazwa w katalogu

**mv plik1 plik2** - zmiana nazwy (pliku lub katalogu)

(muszą być w katalogu bieżącym)

**mv plik1 ./katalog** - przeniesienie do katalogu podrzędnego

istnieje! w katalogu bieżącym

istnieje!

**Alias** to zdefiniowane przez użytkownika polecenie, które odwołuje się do używanych w systemie poleceń wraz z odpowiednimi parametrami.

Polecenie **alias** bez parametrów wyświetla listę zdefiniowanych aliasów. Tym samym poleceniem definiuje się także nowe aliasy.

\$ **alias pokaz='ls -l'**  
wywołanie: \$ **pokaz**

\$ **alias lc='ls -l \*.c'** – polecenie **lc** może być przydatne przy listowaniu plików źródłowych w języku C z danego katalogu

Aby usunąć zdefiniowany alias, należy użyć polecenia **unalias**

\$ **unalias pokaz**

# Ilość miejsca na partycjach

- \$ **df**
- \$ **df -m** – podaje informacje w MB
- \$ **du plik** – ustala ile miejsca zajmuje plik
- \$ **du -a katalog** – pokazuje ilość zajętego miejsca w katalogu wraz z jego elementami
- \$ **du -s -b -c katalog** – pokazuje rozmiar każdego katalogu z osobna bez elementów w nich zawartych
- \$ **du -s -b** – wypisuje całkowitą objętość danego elementu bez zbędnych informacji

## Wyszukiwanie plików i katalogów, polecenia:

- **whereis**,
- **find**,
- **which**

\$ **whereis nazwa\_pliku** – podaje ścieżki dostępu do plików

\$ **which nazwa\_pliku** – podaje ścieżkę dostępu do pliku, który jest wykonywany po wydaniu polecenia

Polecenie **find** pozwala na wyszukiwanie plików wg różnych kryteriów

\$ **find** katalog\_startowy\_szukania **opcje** **kryterium**

przykładowe opcje:

**-name** – szukanie wg nazwy

**-type** – szukanie wg typu;

Wymagany jest jednorazowy argument, którym jest jeden ze znaków:

**d** – katalog

**f** – plik zwykły

**l** – link symboliczny

dostępne opcje (cd.):

- size** – np. **-size +100c** – szukane są pliki o rozmiarze większym niż 100 znaków,  
natomiast **-size -100w** – szukane są pliki o rozmiarze mniejszym niż 100 słów
- mtime** – szukanie wg liczby dni, jakie minęły od ostatniej modyfikacji, np. **-mtime +3** – szuka plików modyfikowanych więcej niż 3 dni temu
- atime** – szukanie wg liczby dni od ostatniego dostępu
- user** – szukanie plików, których właścicielem jest użytkownik
- perm** – szukanie wg praw dostępu np. **-perm 100** pozwala na odnalezienie plików, które mają ustawione przynajmniej prawo x dla właściciela
- never plik1** – szukanie plików modyfikowanych później niż plik plik1



Przy poszukiwaniu można korzystać również z operatorów logicznych OR (-o), NOT (!), AND (-a).

\$ **find ! -name 'agh'** – szuka pliki mające nazwy różne od agh

\$ **find -name 'agh' -o -type d-** pozwala na wyszukanie plików o nazwie agh lub typie d

Kiedy kilka opcji zostanie podanych w wierszu poleceń, tworzą one operację AND

\$ **find / -name 'agh' >p1** – rozpoczyna poszukiwania od głównego katalogu i szuka plików o nazwie agh, a następnie zapisuje rezultat poszukiwań w pliku p1

# Przeglądanie zawartości plików

\$ **cat** **opcje** **nazwa\_pliku**

dostępne opcje:

- b** numerowanie niepustych wierszy
- n** numerowanie wszystkich wierszy

Polecenia **more** i **less** służą do przeglądania pliku po jednej stronie (użyteczne dla dużych plików)



## \$ **more** **opcje** **nazwa\_pliku**

dostępne opcje:

**+#** - rozpoczęcie wyświetlania od wiersza nr #

**-s** – zakaz wyświetlania kilku sąsiednich pustych wierszy

**Enter** przewija o jeden wiersz w dół.

**Spacja** – przejście do następnej strony.

**Klawisz b** – cofnięcie o jeden ekran.

**Klawisz g** – wyjście z polecenia more.

## \$ **less** **nazwa\_pliku**

Polecenie bardzo podobne do more; wyświetla treść pliku i umożliwia przeglądanie za pomocą strzałek

\$ **wc** **opcje** **plik** – powoduje zliczanie liczby znaków, słów, wierszy w pliku tekstowym

dostępne opcje:

- c** – tylko liczba znaków
- l** – tylko liczba wierszy
- w** – tylko liczba słów

## Podstawowe elementy wyrażeń regularnych:

- . -dowolny pojedynczy znak;
- \$ -dopasuj poprzedzające wyrażenie do końca wiersza;
- ^ -dopasuj występujące po operatorze wyrażenie do początku wiersza;
- \* -dopasuj zero lub więcej wyrażeń znaku poprzedzający operator;
- \ -oznacza ominięcie specjalnego znaczenia znaku np: "\\*";
- [ ] -dopasuj dowolny znak w nawiasie;
- [-] -dopasuj dowolny znak z przedziału [0-9];
- [^] -dopasuj znak, który nie znajduje się w nawiasie;

**ls -l | grep student** - wyświetlenie zawartości tylko tych pozycji katalogu, gdzie znajduje się słowo "student" (czyli np. będących własnością studenta, posiadających słowo "student" w nazwie itp).

**cat zrodlo.c | grep include** - wyświetli wszystkie linie pliku zrodlo.c, zawierające ciąg "include".

## Rodzaje wieloznaczników:

W większości przypadków używane są dwa wieloznaczniki:

**wieloznacznik lokalny** – zastępujący pojedyncze wystąpienie dowolnego znaku (do jego oznaczania najczęściej stosowany jest znak zapytania (?)),

**wieloznacznik ogólny** – zastępujący dowolną liczbę dowolnych znaków (do jego oznaczania najczęściej stosowany jest znak gwiazdki (\*)).

## Przykłady:

**grep 'Ala' plik** -znajduje wyraz 'Ala' w pliku;

**grep 'A[lg]a' plik** -znajduje wyraz 'Ala' lub 'Aga';

**grep 'A.a' plik** -znajduje wyrazy takie jak 'Ala' 'Aga' itp;

**grep '^Ala' plik** - znajduje wyraz 'Ala' na początku wiersza;

**grep 'Go\*gle' plik** - znajduje wyraz 'Gogle', 'Google' itd;

**grep '[0-9]'** - znajduje dowolny ciąg znaków z zakresu od 0 do 9;

# Uzyskiwanie informacji o sprzęcie

\$ **arch** – wyświetla informacje o architekturze komputera  
(rodzaj zastosowanego procesora)

\$ **uname** – informacje o wersji oprogramowania  
dostępne opcje:

\$ **uname -a** – wszystkie informacje o systemie

\$ **uname -m** – wyświetla tylko architekturę maszyny

\$ **uname -n** – nazwa hosta komputera

\$ **uname -p** – wypisuje typ procesora

\$ **uname -r** – informacje o wersji jądra linuxa

\$ **uname -s** – wypisuje nazwę systemu operacyjnego

\$ **uname -v** – wersja systemu operacyjnego

\$ **adduser nazwa\_uzytkownika** – dodawanie nowego użytkownika

**-e** – data ważności, po której stanie się ono nieaktywne

# **adduser -e 01/01/09 adam**

**-f** – dni nieaktywności, określa liczbę dni po wygaśnięciu ważności hasła, jaka pozostała do wygaśnięcia ważności konta

# **adduser -f 4 janek**

**-g** – grupa początkowa, definiuje ją dla użytkownika, jeżeli nie istnieje to trzeba ją założyć

# **adduser -g grupa1 maja**

**-G** – określa listę grup do których będzie przypisany użytkownik

# **adduser -G gr1 gr2 wiola**

**-s** – określamy powłokę dla użytkownika

# **adduser -s /bin/sh ola**

- # **userdel jarek** – usuwanie konta z systemu bez usuwania plików pozostawionych przez użytkownika w innych lokalizacjach (trzeba je usunąć „ręcznie”)
- # **userdel -r jarek** – usuwa katalog domowy użytkownika wraz z plikami

- \$ **finger nazwa\_uzytkownika** – informacje o konkretnym użytkowniku (używanej powłóce, miejscu przydzielonym na dysku, czasie ostatniego logowania)
- \$ **w** – kto jest obecnie zalogowany
- \$ **free** – informacje o pamięci systemowej  
(**-b** w bajtach, **-k** w kilobajtach, **-m** w megabajtach)



## Grupy

Grupa to sposób na przydzielenie użytkownikowi dostępu do danego programu lub innego zasobu komputera

# **groupadd grupa1** – dodawanie nowej grupy

# **groupdel grupa1** – usuwanie grupy

d rwx rwx rwx

właściciel

grupa

pozostali

## Prawa dostępu

### prawa do pliku

r – do czytania

w – do modyfikacji

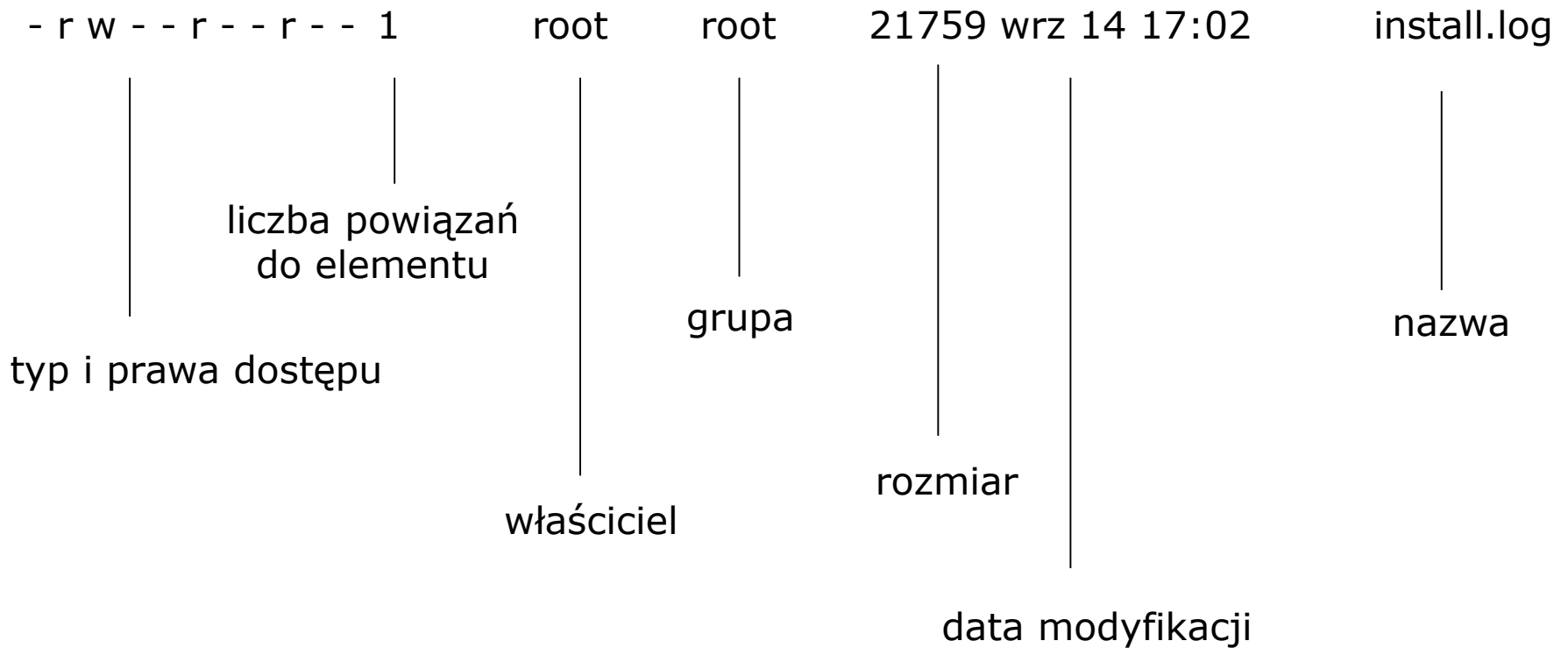
x – do uruchomienia

### prawa do katalogu

r – do przeszukiwania

w – do zmiany zawartości

x – do wejścia



## Dane o plikach i katalogach

Punkty przysługujące poszczególnym kategoriom użytkowników należy złożyć razem, np. `rw-r--r--`

prawo dostępu właściciela `rw-`  $4+2+0=6$

prawa dostępu grupy `r--`  $4+0+0=4$

prawa innych użytkowników `r--`  $4+0+0=4$

\$ **`chmod 644 plik1`**

\$ **`chmod -c 666 plik1`** – wyświetla informacje o zmienionych plikach

# **man polecenie** – wyświetla instrukcję do wskazanego polecenia. Instrukcje pochodzą z dokumentacji systemu Unix.