

Wstęp do systemu Linux

v.1.1

Plan

- Wprowadzenie do systemów Unix
 - Poruszanie się po konsoli, system plików
 - Użytkownicy
 - Dostęp do plików
- Instalowanie nowych programów
 - Tematy zaawansowane
 - Różnorodność w systemach Linux

Wprowadzenie do systemów Unix

Wprowadzenie do systemów Unix

Krótko o Linuksie

- Linux to rodzina uniksopodobnych systemów operacyjnych.
- Linux jest jednym z przykładów wolnego i otwartego oprogramowania.
- Jego kod źródłowy może być dowolnie wykorzystywany, modyfikowany i rozpowszechniany.

Co to jest shell?

- Shell jest najniższą powłoką interfejsu użytkownika typową dla systemów uniksowych.
- Jest to część systemu odpowiedzialna za podstawową interakcję z użytkownikiem.
- Każdy z shelli musi implementować podstawową liczbę komend wymaganych przez standard.
- Najczęściej każdy z nich usprawnia standard przez rozszerzenie liczby poleceń.

Przydatne skróty

CTRL + C	Przerywanie pracy procesu
CTRL + D	Wysyła sygnał EOF (end-of-file)
CTRL + R	Przeszukuje historię pod względem podanych liter
CTRL + Z	Zatrzymanie procesu
CTRL + A	Przeniesienie kursora na początek linii

Wprowadzenie do systemów Unix

Historia poleceń

Powłoka shell zapamiętuje ostatnio używane komendy (domyślnie – 1000):

- **history** – pokazuje listę używanych komend,
- **!!** – wykonuje ostatnią komendę,
- **!-3** – wykonuje trzecią komendę od końca z listy,
- **!5** - wykonuje piątą komendę z listy,
- **!grep** – wywołuje ostatnią komendę zaczynającą się od **grep**,

W nowszych shellach do wywołania historii poleceń służy skrót klawiszowy: **CTRL + R**.

Edytory tekstu

- **Gedit** – podstawowy (zainstalowany od początku) edytor tekstu w Ubuntu. Ma wspomaganie dla systemu kodowania UTF-8.
- **Vi** – podstawowy edytor tekstu w konsoli. Nieporęczny, problematyczny i trudny. Warto jednak nauczyć się jego obsługi, jeżeli pracujemy przez SSH. Niezastąpiony przy pracy z wielkimi plikami.
- **Geany** – słynny multiplatformowy edytor tekstu. Bogata liczba opcji czyni go jednym z lepszych edytorów dla programistów.

Edytor Vi

Podstawy użytkowania Vi

Vi działa w dwóch trybach:

- komend – tryb, w którym wpisujemy komendy programu (takie jak: zapisz plik, zamknij program itp.). Żeby z niego przejść do trybu edycji należy wcisnąć klawisz **I** (od słowa „insert”).
- edycji – tryb, w którym mamy możliwość edycji pliku. Żeby z niego przejść do trybu komend wciskamy klawisz **ESC**.

Podstawowe komendy Vi

:30	Przesuwa kursor do wskazanej linii
/<ciąg_znaków>	Wyszukuje dany napis, np. /anything
? <ciąg_znaków>	Wyszukuje dany napis wstecz (od końca pliku)
n	Znajduje następne wystąpienie danego wyszukiwania
N	Znajduje poprzednie wystąpienie danego wyszukiwania
:e <nazwa_pliku>	Otwiera nowy plik o podanej nazwie
:w	Zapisuje plik
:w!	Zapisuje plik, nadpisując pozwolenia tego pliku (zdejmuje read-only)
:w <nazwa_pliku>	Zapisuje do nowego pliku o podanej nazwie
:q	Wychodzi z programu

**Poruszanie się
po konsoli,
system plików**

Struktura katalogów w systemie

/	Główny katalog w systemie (wszystkie katalogi są podkatalogami /)
/dev	Katalog, w którym znajdują się wszystkie urządzenia
/proc	Katalog wymiany danych komunikacji międzyprocesowej, zawiera też szczególne informacje na temat systemu. Nie zawiera w sobie żadnego „realnego” pliku
/etc	Katalog zawierający pliki konfiguracyjne
/sbin	Katalog zawierający podstawowe pliki binarne potrzebne do działania systemu
/lib	Katalog zawierający biblioteki zainstalowane w systemie
/mnt	Katalog, w którym montowane są wszystkie dyski
/bin	Katalog zawierający programy

/etc	Pliki używane przez podsystemy Uniksa (np. bazy danych)
/etc/init.d	Katalog zawierający skrypty uruchamiane podczas startu systemu
/etc/profile.d	Katalog zawierający skrypty uruchamiane przy logowaniu danego użytkownika ()
/home	Katalog domowy użytkownika
/root	Katalog domowy użytkownika root (głównego użytkownika systemu)
/tmp	Katalog zawierający pliki chwilowe potrzebne do działania programów i systemu
/usr	Katalog zawierający pliki wykonywalne programów, kod źródłowy, biblioteki i dokumentacje

Podstawowe komendy – pliki i katalogi

ls	Wyświetla wszystkie pliki	ls -a ls -l	Wyświetla także pliki ukryte Wyświetla dodatkowe informacje
mkdir dirname	Tworzy katalog		
cd dirname	Przechodzi do wskazanego katalogu	cd . cd .. cd ~	Obecny katalog Katalog bezpośrednio wyżej Katalog domowy
pwd	Wyświetla ścieżkę do katalogu, w którym się znajdujemy		
cp file1 file2	Kopiuje file1 na miejsce file2		
mv file1 file2	Przenosi file1 na miejsce file2		

Podstawowe komendy – pliki i katalogi

rm file	Usuwa plik	rm -r	Usuwa także katalogi
rmdir dirname	Usuwa katalog		
cat file	Wyświetla wskazany plik		
less file	Wyświetla plik strona po stronie		
head file	Wyświetla pierwsze 10 linii pliku	head -n	Wyświetla n linii
tail file	Wyświetla ostatnie 10 linii pliku	tail -n	Wyświetla n linii
wc file	Podaje liczbę słów, znaków, linii lub bajtów w pliku (lub potoku)	wc -c wc -l wc -w wc -m	liczba bitów, liczba bitów, liczba słów, liczba znaków
touch file	Tworzy plik o podanej nazwie		

Podręczniki systemowe

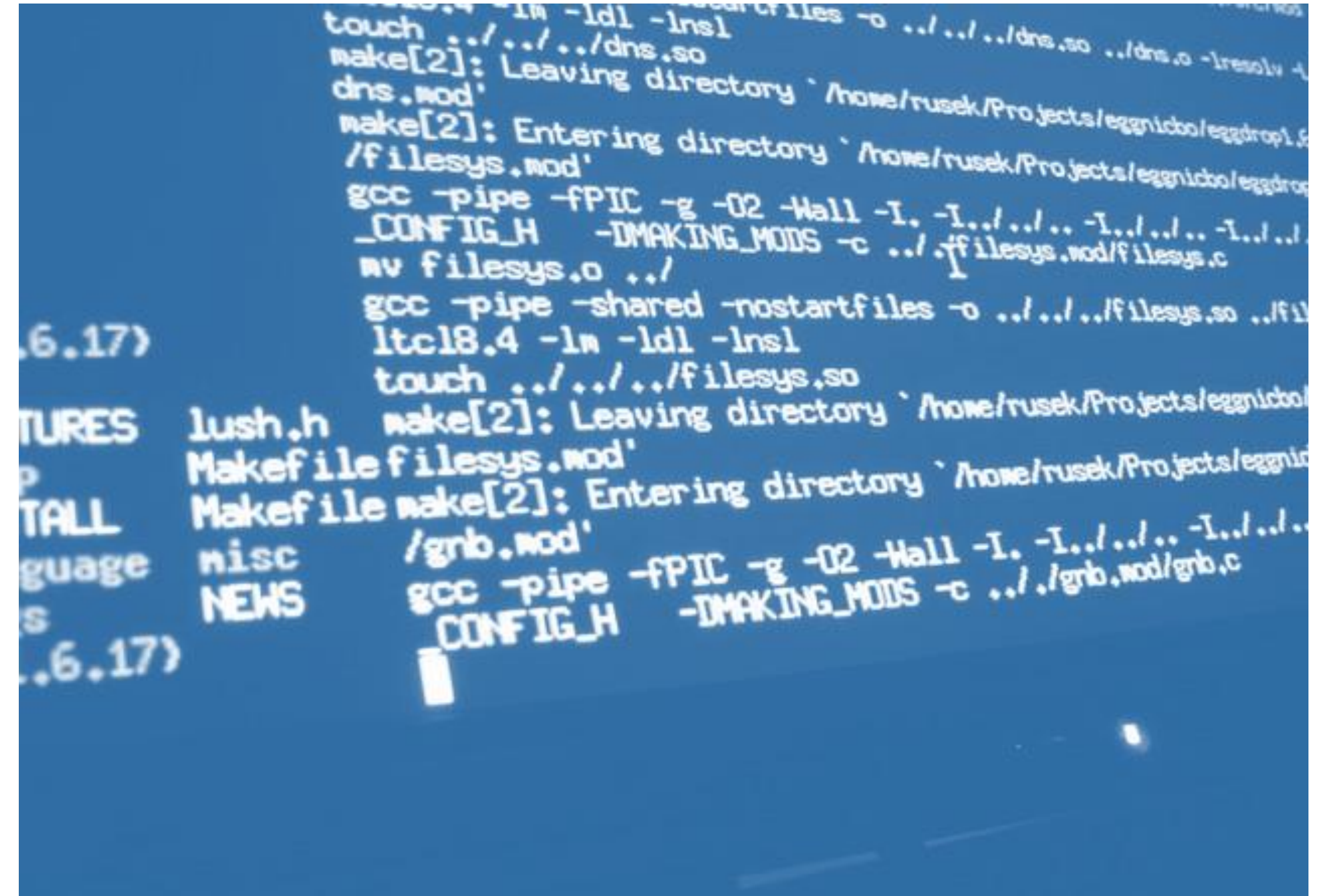
man <nazwa_komendy> – otwiera podręcznik pomocy danej komendy np. **man** ls.

Komenda z jednym z tych parametrów wyświetla dodatkowe instrukcje:

<nazwa_komendy> **--help**

<nazwa_komendy> **-h**

apropos com – wyświetla wszystkie komendy mające słowo **com** w nagłówku swojego podręcznika.



Podstawowe komendy – grep

- Grep służy do wyszukiwania danego ciągu znaków w podanych plikach. Podstawowym użyciem jest: `grep "wyszukiwana fraza" plik`.
- W takim przypadku wynikiem są wszystkie linie zawierające daną frazę w podanym pliku.

Przykład

`grep -i "lorem" readme.txt`

Podstawowe opcje komendy grep

-i	Wyszukuje, nie zważając na wielkość znaków
-w	Wyszukuje tylko pełne słowo
-A <n>	Pokazuje n linii po wyszukiwanym słowie
-B <n>	Pokazuje n linii przed wyszukiwanym słowem
-r	Wyszukuje rekursywnie we wszystkich plikach podanego katalogu
-v	Wyszukuje wszystkie linijki niezawierające podanego słowa
-c	Podaje liczbę wystąpień wyszukiwanego słowa
-l	Wypisuje nazwy plików, w których znalazł dane słowo
-n	Dodaje numer linii, w której znalazł słowo

Podstawowe komendy

find

Komenda wyszukująca pliki to **find**.
Jej ogólna forma to:

```
find <katalog startowy> <kryteria  
wyszukania i akcje>
```

Przykład

```
find . -name "pattern" -print
```



Podstawowe opcje komendy find

-atime n	Plik, który został utworzony n dni temu, np. +7 – utworzony dawniej niż siedem dni temu
-mtime n	Plik, który został zmodyfikowany n dni temu, np. -5 – zmodyfikowany nie później niż 5 dni temu
-size n	Plik ma n bloków wielkości (blok to 512 bajtów), np. +100 – plik większy niż 100 bloków = 50 KB
-type f	Wyszukuje po typie pliku, np. f = plik tekstowy (w przykładzie), d = katalog
-name nam	Nazwa pliku to nam

-user usr	Nazwa właściciela pliku to usr
-group grp	Właściciel pliku należy do grupy grp
-perm p	Tryb dostępu pliku to p (gdzie p to liczba)
-print	Wyświetla ścieżkę do pliku
-exec cmd	Wykonuje komendę cmd na pliku

Potok

Potok

Potok (pipe) – jeden z mechanizmów komunikacji międzyprocesowej, umożliwiający wymianę danych pomiędzy dwoma procesami. Odbywa się to najczęściej przez połączenie STDOUT z STDIN innego procesu, na przykład:

```
ps aux | less
```

```
cat plik | grep -i a
```

command > file	Przekierowuje STDOUT z komendy command do pliku file (nadpisując go)
command >> file	Przekierowuje STDOUT z komendy command do pliku file (rozszerzając go)
command < file	Przekierowuje STDIN z pliku file do komendy command pliku file
cat file1 file2 > file0	Skleja file1 i file2 wynik, zapisując do file0



Użytkownicy

Rodzaje użytkowników w systemach Unix

Oto trzy główne typy użytkowników:

- **root** – tak zwany **superuser** – ma całkowity dostęp do maszyny, może wywoływać każdą komendę,
- **konta systemowe** – potrzebne do działania systemu i krytycznych dla niego procesów,
- **konta użytkowników** – konto normalnego użytkownika.

Podstawowe komendy

adduser username	Dodaje użytkownika do systemu	-d homedir – wskazuje na (już istniejący) katalog domowy -g groupname – dodaje do danej grupy podczas tworzenia -m – tworzy nowy katalog domowy -l – zmienia nazwę użytkownika (tylko dla usermod)
usermod username	Zmienia atrybuty użytkownika	
passwd username	Zmienia hasło użytkownika	
deluser username	Usuwa użytkownika	-r – niszczy katalog domowy danego użytkownika

Podstawowe komendy

Grupy

addgroup groupname	Dodaje grupę do systemu	-g ID – numer ID grupy,
groupmod groupname	Zmienia opcje grupy	-o – daje możliwość użycia zajętego już numeru ID, -r – dodaje konto systemowe do grupy, -f – opcja ta powoduje, że funkcja zwróci success, jeżeli grupa już istnieje, -n – zmienia nazwę grupy (tylko groupmod).
delgroup groupname	Usuwa grupę	

SUDO

- **sudo nazwa_komendy** – wywołuje daną komendę na poziomie administratora systemu (podobne możliwości jak **root**).
- **sudo su** – otwiera nową powłokę, w której działamy jako admin.
- **visudo** – pozwala na bezpieczną konfigurację pliku sudoers (oznaczającego, kto ma prawa do używania komendy sudo).



Dostęp do plików

Prawa dostępu

Po wpisaniu komendy:

ls -lg

– przykładowy output jest następujący:

**drwxr-xr-x 1 Agata 197610 0 mar 18 13:19
katalog_z_obrazkami/**

- Pierwszy symbol (w tym zapisie **drwxr-xr-x**) oznacza, czy dany element jest katalogiem czy nie (czyli d oznacza katalog, plik jest określany kreską -)
- Następne 9, to opis praw dostępu.
- Dalej jest suma kontrolna, nazwa grupy, do której należy plik, wielkość, data utworzenia i nazwa.

Opis rwx

- Pierwsze trzy znaki oznaczają możliwości dostępu dla właśnie zalogowanego użytkownika (r – read, w – write, x – execute)
- Dalsze trzy oznaczają dostęp dla grupy, do której należy dany plik.
- Ostatnie trzy – prawa dostępu dla wszystkich innych.

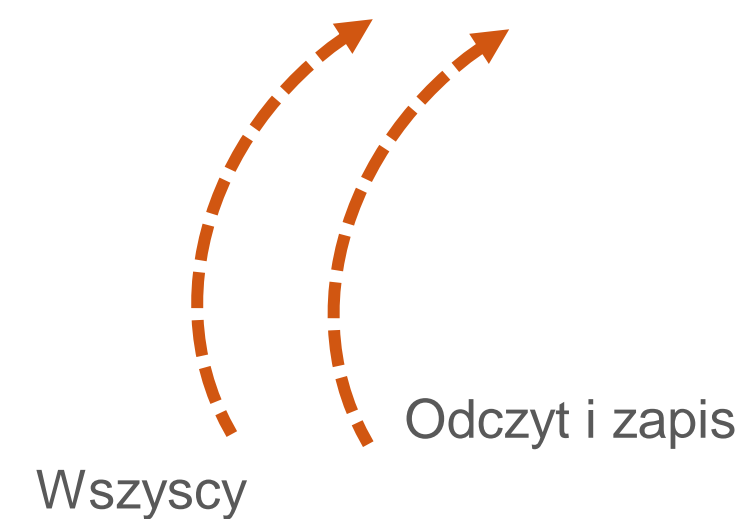
-rwxrw-r--

Zmiana praw dostępu do pliku

chmod – komenda zmieniająca uprawnienia dostępu do pliku.

Przykład

chmod a=rw file.txt



U	Użytkownik
G	Grupa
O	Inni
a	Wszyscy (to samo co połączenie u , g , o)
r	Odczyt
w	Zapis (i usunięcie)
x	Uruchomienie (w przypadku katalogu dostęp)
+	Dodanie uprawnień
-	Zabranie uprawnień

Zmiana grupy, do której należy plik

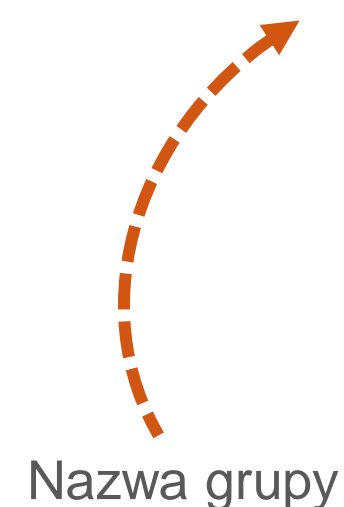
chgrp

chgrp – komenda zmieniająca grupę pliku tylko do takiej grupy, do której użytkownik sam należy.

chgrp nazwa_grupy plik1 plik2

Przykład

chgrp CodersLab cwiczenie1.txt



Nazwa grupy

chown

chown – komenda służąca do zmiany właściciela pliku (co zmienia też grupę). Może być wywoływana tylko przez administratora systemu (poprzez **sudo**).

chown :nazwa_grupy plik1 plik2



Nowy właściciel pliku

Nazwa grupy, do której przypisujemy plik

Nazwa pliku, plików



Instalowanie nowych programów

Instalowanie menedżera pakietów na MacOS

- System operacyjny MacOS nie ma domyślnie menedżera pakietów.
- Aby zainstalować najpopularniejszego managera wejdź na stronę <http://brew.sh> i wpisać w terminal podaną tam komendę instalacyjną.

Uwaga! Brew nigdy nie powinno być używane z sudo!



Mac OS

Zarządzanie pakietami

Zarządzanie pakietami (apt)

Aktualizowanie listy pakietów:

- Linux
sudo apt-get update
- MacOS
brew upgrade

Instalacja pakietu:

- Linux
sudo apt-get install nazwa_pakietu
- MacOS
brew install nazwa_pakietu

Kasowanie pakietów:

- Linux:
sudo apt-get remove nazwa_pakietu
- MacOS:
brew uninstall nazwa_pakietu

Kasowanie pakietu z zależnościami:

- Linux:
sudo apt-get --purge remove nazwa_pakietu
- MacOS:
brew uninstall nazwa_pakietu

Zarządzanie pakietami

Pobieranie kodów źródłowych:

- Linux
sudo apt-get source nazwa_pakietu

- MacOS – niezaimplementowane

Wyszukiwanie pakietów:

- Linux
sudo apt-cache search nazwa_pakietu
- MacOS
brew search nazwa_pakietu

Aktualizowanie wszystkich pakietów:
sudo apt-get upgrade

Aktualizowanie dystrybucji:
sudo apt-get dist-upgrade

Kasowanie wszystkich pobranych plików:
sudo apt-get clean

Zarządzanie pakietami

Zarządzanie pakietami (dpkg)

- Aktualizowanie wszystkich pakietów:
sudo apt-get upgrade
- Aktualizowanie dystrybucji:
sudo apt-get dist-upgrade
- Kasowanie wszystkich pobranych plików:
sudo apt-get clean

Polecenie **dpkg** służy do instalacji pobranych plików .deb.

- Instalacja pakietu:
sudo dpkg nazwa_pakietu
- Kasowanie pakietu:
sudo dpkg -r nazwa_pakietu



Tematy zaawansowane

Procesy

- **ps** – komenda wypisująca wszystkie procesy.
- Użyteczna w połączeniu z **grep**, poniższa komenda pokaże wszystkie procesy, które w nazwie mają „chrome”:

ps aux | grep chrome

- **pstree** – pokazuje procesy (tylko te należące do użytkownika) w formie drzewa procesów.

Opcje komendy ps

-a	Pokazuje procesy innych użytkowników
-e	Pokazuje rozszerzone informacje
-u	Pokazuje dodatkowe informacje (jak opcja -f)
-x	Pokazuje informacje o procesach nieznajdujących się w terminalu

Procesy

Informacje wyświetlane przez PS

UID	ID użytkownika, który stworzył proces
PID	ID procesu
PPID	ID procesu rodzica
C	Procent CPU, jaki pochłania proces
STIME	Czas startu procesu
TTY	Terminal, na którym działa proces
TIME	Czas CPU, jaki proces zużył
CMD	Komenda, jaka wystartowała proces

Procesy

Typy procesów

- **Zombie** – proces, który nadal jest widoczny w tabeli procesów, choć się skończył. Stan taki może nastąpić, jeżeli proces rodzic został zamknięty niepoprawnie. Często opisywany też jako **defunct**.
- **Orphan** – działający proces, którego rodzic został zniszczony. Proces taki może cały czas poprawnie się zamknąć.
- **Deamon** – proces systemowy działający w tle bez podpiętego terminala. Zazwyczaj celem demona jest ciągle lub okresowe powtarzanie jakiegoś działania.

Niszczanie procesów

- **kill [sygnał] [PID]** – komenda wysyłająca sygnał do procesu. Sygnały niszczące (zabijające) procesy:
-SIGTERM (-15)
-SIGKILL (-9)
- **killall [nazwa-procesu]** – Wysyła sygnał do wszystkich procesów o danej nazwie.
- Obie komendy wyślą **SIGTERM**, jeżeli nie zostanie podany żaden sygnał.
- Żeby zabić proces **zombie** najczęściej trzeba zabić proces jego rodzica (**PPID**).

Praca ze zdalną konsolą

Praca ze zdalną konsolą

- SSH – skrót od secure shell. Protokół pozwalający na bezpieczne zalogowanie się screen do komputera przez sieć.
- Logujemy się poprzez komendę:
ssh user@host.pl
- Przydatne komendy podczas używania SSH:
w – lista zalogowanych osób,
whoami – pokazuje login aktualnie zalogowanego użytkownika,
uptime – pokazuje, ile czasu upłynęło od startu systemu.

Komenda screen

- **screen** – program pozwalający na tworzenie wirtualnych sesji. Sesje te działają do czasu wyłączenia systemu lub ręcznego ich zamknięcia. Bardzo przydatne przy uruchamianiu skryptów przez SSH.
- **screen -S nazwa_sesji** – tworzy sesję o podanej nazwie.
- **screen -d -R nazwa_sesji** – przywraca sesję.
- **CTRL+A+D** – odłącza sesję (nie zamykając jej).
- **CTRL+A+K** – zamyka sesję.

Harmonogram zadań

- **cron** – demon (proces działający w tle), którego praca polega na okresowym wywoływaniu innych programów.
- **crontab** – tabela zadań, które **cron** ma uruchamiać, z dokładnym określeniem czasu, w którym mają być uruchomione.

Opcje

-e	Edycja.
-v	Wyświetlenie czasu ostatniej edycji.
-l	Wyświetlenie.
-r	Usunięcie całego pliku crontab.

Przykładowy wygląd pliku crontab

Przykład

- Aby dodać zadanie, które będzie uruchomione co określony czas, musimy dodać */<odstęp czasu> w odpowiednim polu.

- */5 * * * * /backup.sh
Uruchomi skrypt co pięć minut.

```
# For details see man 4 crontabs

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name  command to be executed
```


Zmienne systemowe

- **Zmienne środowiska shell** – zmienne krótkoterminowe, czyszczone pod koniec działania powłoki.
- **Zmienne systemowe** – zmienne długoterminowe, zapamiętywane między sesjami użytkownika.
- Wypisanie zmiennej:
echo \$<nazwa_zmiennej>
- Nastawienie zmiennej:
set <nazwa_zmiennej> = wartość

Zmienne systemowe		Zmienne shella	
USER	Nazwa zalogowanego użytkownika	cwd	Ścieżka, w której się znajdujesz
HOME	Ścieżka do katalogu home	home	Ścieżka katalogu domowego
HOST	Nazwa komputera	path	Katalogi, w których shell szuka programów do wywołania
ARCH	Architektura procesora		
DISPLAY	Nazwa środowiska graficznego		
PATH	Katalogi, w których shell szuka programów do wywołania		

Symlinki i hardlinki

Tworzenie

- **symlink** – wskaźnik na plik znajdujący się w innym miejscu. Jeżeli zmienimy nazwę pliku lub przeniesiemy go, **symlink** zostanie zepsuty. Jeżeli plik zostanie podmieniony, **symlink** zacznie wskazywać na nowy plik.
- **hardlink** – wskaźnik na docelowe miejsce na dysku (**inode**). W chwili przeniesienia pliku **hardlink** będzie poprawnie na niego wskazywał. Może być utworzony tylko na tym samym systemie plików.

- **Symlinki:**
In /root/file1 /root/file2
- **Hardlinki:**
In -s /root/file1 /root/file2

Różnorodność w systemach Linux

Najpopularniejsze wersje Linuksa

➤ Ubuntu

- jedna z najpopularniejszych dystrybucji Linuksa,
- ma wiele własnych dystrybucji.

➤ Linux Mint

- user experience bardzo podobny do systemu Windows,
- system działający na zasadzie out of the box.

➤ Debian

- jedna ze starszych dystrybucji,
- służył jako baza m.in. dla Ubuntu,
- czysty system operacyjny.

➤ Fedora

- system wprowadzający najwięcej zmian, ciągle dodający najnowsze udogodnienia,
- bardziej problematyczna instalacja systemu, mniejsza stabilność.

➤ OpenSUSE

- alternatywa dla Mint, Ubuntu i podobnych systemów,
- łatwy w instalacji i użytkowaniu.

➤ Arch

- system dla zaawansowanych użytkowników,
- Daje możliwość stworzenia całkowicie spersonalizowanego systemu.

Najpopularniejsze typy shelli

Bourne Shell (sh)

- Dostępna na każdym systemie typu Unix (wyznacza standard).
- Druga powłoka używana w systemach Unix (stworzona w 1977 roku).
- Główne ograniczenie to niemożliwość działania na liczbach całkowitych bez tworzenia nowego procesu.
- Można go zidentyfikować podczas używania po znaku \$ znajdującym się na początku linii.

Bash

- Akronim od Bourne-Again Shell.
- Domyślna powłoka w większości systemów typu Linux oraz w systemie Mac OS X (wersie 10.3+).
- Pozwala na pracę w trybie konwersacyjnym (interaktywne wprowadzanie poleceń) i wsadowym (poprzez skrypty).
- Rozszerza standard **sh** np. przez:
 - działania na liczbach całkowitych,
 - przekierowywanie wejścia i wyjścia,
 - wyrażenia regularne (Bash 3.0+).

Najpopularniejsze typy shelli

Z shell (zsh)

- Potężne rozwinięcie standardu **sh** dla zaawansowanych użytkowników zawierające m.in.:
 - programowalne autouzupełnianie komend,
 - współdzielenie historii komend pomiędzy działającymi powłokami,
 - rozbudowane wyszukiwanie plików (nieopierające się na programach typu **find**),
 - autokorektę,
 - całkowitą kompatybilność z **sh** (może się podszywać pod powłokę **sh**).

C shell (csh)

- Powłoka stworzona dla systemu BSD.
- Główna zmiana polega na stworzeniu języka podobnego do C jako języka głównego powłoki.
- Pomimo dodania wielu usprawnień do standardu powłoka nie przyjęła się i jest uważana za problematyczną.

Najlepsze emulatory terminalu

Terminator

- Zaawansowany i uznawany za jeden z najlepszych emulatorów.
- Główne jego funkcjonalności to:
 - różne schematy kolorystyczne (także user defined),
 - możliwość doinstalowania różnych pluginów,
 - dodatkowe skróty klawiszowe dla najczęstszych komend,
 - dzielenie okna na pomniejsze wirtualne terminale i możliwość zmiany ich wielkości.

Guake

- Emulator całkowicie napisany w Pythonie.
- Jako jeden z pierwszych wprowadził ukrywanie emulatora pod górnym paskiem systemowym (bazowane na emulatorach z gier FPS).
- Stworzony dla środowiska graficznego GNOME.

Najlepsze emulatory terminalu

Yakuake

- Emulator podobny do Guake, przeznaczony dla systemów opartych na środowisku graficznym KDE.
- Główne cechy:
 - konfigurowalna wielkość i animacja opadania,
 - interfejs tabelkowy.





**GADAĆ JEST ŁATWO.
POKAŹCIE MI KOD.**

L. TORVALDS