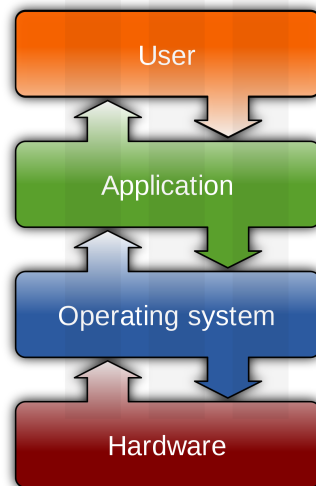


# Podstawy Linux

2 grudnia 2018

## 1 Wstęp

Linux - jądro (kernel) systemu operacyjnego - interfejs\* pomiędzy sprzętem (hardware\*) i aplikacjami - odpowiada za kolejność wykonywania procesów, moduły bezpieczeństwa, zarządzanie ruchem sieciowym, sterowniki sprzętowe. Systemy operacyjne oparte na jądrze Linux korzystają z bibliotek\* GNU, tzw. systemy GNU/Linux. Różne wersje Linuxa nazywamy dystrybucjami\*, np. Ubuntu, Mint, Scientific Linux, Manjaro itd.



Rysunek 1: [https://en.wikipedia.org/wiki/Operating\\_system](https://en.wikipedia.org/wiki/Operating_system)

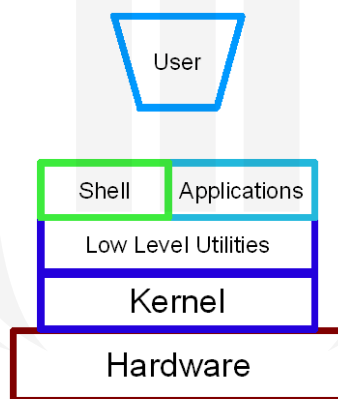
- Interfejs - połączenie dwustronne, które można traktować jako tłumacza - udostępnia funkcje jednej strony w języku drugiej. Np. pilot jest interfejsem między nami a telewizorem - przyciski są opisane obrazkami i cyframi, które do telewizora przesyłane są jako odpowiednie sygnały
- Hardware - Komponenty składające się na komputer np. karta sieciowa, klawiatura, procesor itd.

- Biblioteki - gotowe podstawowe programy tzw. podprogramy, zbiory funkcjonalności, możliwe do wykorzystania w różnych programach. Przykłady:
  - biblioteka odpowiadająca za rysowanie i obsługę elementów strony internetowej - przyciski, suwaki, wyświetlanie obrazów, hiperłącza itd.
  - biblioteka z funkcjami matematycznymi
- Dystrybucje Linux - wersje systemów operacyjnych odpowiadających na inne potrzeby użytkowników np. ogólnego przeznaczenia lub o specjalistycznych zastosowaniach

Systemy Linux'owe połączone są z darmowymi repozytoriami (serwery z oprogramowaniem). Zarządzanie repozytoriami, pobieranie, aktualizowanie oraz usuwanie programów możliwe jest za pomocą programu **apt**. Programy te instalują się automatycznie wraz ze wszystkimi potrzebnymi programami (tzw. dependencies - pliki i programy niezbędne do działania danego programu).

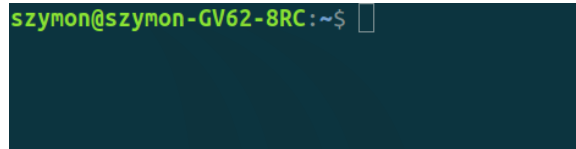
## 2 Shell

Powłoka systemu (Shell - kolejna część systemu) - interfejs użytkownika (user) - program pozwalający na komunikację między systemem a użytkownikiem za pomocą poleceń tekstowych (CLI - Command Line Interface). W systemach Linux taką powłoką jest program bash. W przeszłości był to jedyny sposób na komunikację z systemem, teraz dodatkowo mamy do dyspozycji GUI (Graphical User Interface). W przypadku systemu Windows mamy do czynienia z Windows Shell - GUI (to co widzimy, pasek start, menu start, pulpit) oraz cmd - CLI (odpowiednik shell'a w linux).



Rysunek 2: <http://www.penguintutor.com/linux/basic-shell-reference>

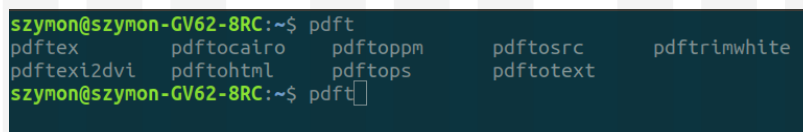
GUI Linux nie różni się zasadniczo od GUI w systemie Windows. Na zajęciach będziemy korzystać tylko z powłoki systemu. Okno Shella jest bardzo proste - prompt. Prompt w wersji podstawowej ma następującą formę: **nazwa\_uzytkownika:aktualna\_sciezka\$**.



Rysunek 3: Przykładowy prompt

## 2.1 Podstawy

- Przycisk *tab* służy do automatycznego uzupełniania wpisywanej komendy. Uzupełnia nazwy komend, plików oraz katalogów. Gdy istnieje wiele możliwości, pojedyncze wciśnięcie przycisku nic nie da, podwójne wyświetli listę możliwości (podpowiedzi).



Rysunek 4: Dwukrotne wciśnięcie *tab* po wpisaniu *pdft*

- Strzałki w górę oraz w dół powodują przejście po historii wykonywanych poleceń - możliwość ponownego wykonania polecenia lub edycji źle wpisanego polecenia.
- Gdy po wykonaniu komendy nie wyświetli się żaden komunikat - oznacza to, że wszystko poszło zgodnie z planem
- Niektóre czynności wymagają uprawnień administratora, wtedy należy wywołać komendę poprzedzając ją komendą **sudo** (super user do) np. **sudo rm plik.txt**

## 2.2 Pliki i katalogi

Pliki w systemie Linux nie mają rozszerzenia, dodawane jest ono tylko dla wygody i organizacji. Katalogi przechowują informacje o plikach oraz strukturze ich ułożenia. Plik jest jednoznacznie charakteryzowany przez ścieżkę wraz z nazwą pliku np. */home/user/plik*. Symbol */* oddziela nazwy folderów od ich zawartości. Pliki oraz katalogi zaczynające się od znaku *."* są plikami ukrytymi. Katalogi zawsze posiadają wewnątrz 2 katalogi *."* - oznaczającą ten sam katalog oraz *."* - adres katalogu nadrzędnego. Dzięki temu możemy odwoływać się do katalogów ścieżką względną, np. *./katalog1/* oznacza katalog wewnątrz katalogu w którym się znajdujemy, a *../katalog2* oznacza katalog wewnątrz katalogu nadrzędnego. Zazwyczaj pliki użytkownika trzymamy w katalogu domowym, do którego najłatwiej odnieść się poprzez znak *~* np. **ls ~** lub **ls ~/katalog**.

W systemach Linux katalog *"/* jest tzw. *root directory* - katalog w którym zawarte są wszystkie dostępne w systemie katalogi oraz pliki. Najważniejsze podkatalogi:

```
szymon@szymon-GV62-8RC:~/Folder$ ls
InnyFolder Plik
szymon@szymon-GV62-8RC:~/Folder$ ls -a
. .. InnyFolder Plik .ukrytyFolder .ukrytyPlik
szymon@szymon-GV62-8RC:~/Folder$
```

Rysunek 5: `ls` - wypisanie plików i folderów, `ls -a` - wypisanie plików i folderów, także zaczynających się od `.`

```
szymon@szymon-GV62-8RC:~/Folder$ tree
.
├── InnyFolder
│   └── InnyPlik
└── plik

1 directory, 2 files
szymon@szymon-GV62-8RC:~/Folder$ cd InnyFolder/
szymon@szymon-GV62-8RC:~/Folder/InnyFolder$ ls .
InnyPlik
szymon@szymon-GV62-8RC:~/Folder/InnyFolder$ ls ..
InnyFolder plik
szymon@szymon-GV62-8RC:~/Folder/InnyFolder$ cd ..
szymon@szymon-GV62-8RC:~/Folder$
```

Rysunek 6: `tree` - wypisanie struktury plików i folderów zaczynając od aktualnego katalogu

- `/bin` - katalog z plikami wykonywalnymi (executable file) wymaganymi do poprawnej pracy systemu
- `/usr` - katalog z programami użytkownika
- `/home` - katalog roboczy z plikami użytkownika
- `/root` - katalog roboczy administratora (root == super user)

```
szymon@szymon-GV62-8RC:~/Folder$ ls /
bin    dev    initrd.img    lib32    media    proc    sbin    sys    var
boot  etc    initrd.img.old  lib64    mnt     root    snap    tmp    vmlinuz
cdrom  home  lib            lost+found  opt     run     srv     usr    vmlinuz.old
szymon@szymon-GV62-8RC:~/Folder$
```

Rysunek 7: `ls /` - wypisanie zawartości katalogu `/`

## 2.3 Polecenia

Polecenia wykonywane są zazwyczaj z następującą składnią: **nazwa\_polecenia** [**flagi** [**argumenty\_dla\_flag**]] [**argumenty\_dla\_polecenia**]. `[]` oznacza opcjonalność (polecenie może wymagać argumentów lub nie, tak samo flaga).

```
szymon@szymon-GV62-8RC:~/Folder$ head -n 2 plik
With my then if up many. Lain week nay she them her she. Extremity so attending objection
as engrossed gentleman something.
Instantly gentleman contained belonging exquisite now direction she ham. West room at sent
if year.
szymon@szymon-GV62-8RC:~/Folder$
```

Rysunek 8: polecenie **head** wypisuje pierwsze 10 linii pliku. flaga **-n n** zmienia liczbę wypisanych linii na **n**

Większość poleceń jest bardzo dobrze opisanych w manualu basha: *man nazwa\_polecenia*. Jeżeli nie znamy polecenia można przeszukiwać manual wywołując *man -K "szukana fraza"*.

Wstawienie znaku "&" za poleceniem spowoduje wywołanie polecenia w tle - gdy polecenie jest wykonywane możemy dalej korzystać z konsoli.

## 2.4 Podstawowe polecenia w Bash'u

- **pwd** - wypisuje na ekran aktualny katalog
- **ls** - wypisuje liste plików oraz podkatalogów danego katalogu, przydatne opcje -l, -h, -a. Domyślnie dotyczy aktualnego katalogu (.).
- **cd** - przechodzi do podanego katalogu. Bez podania katalogu przejdzie do katalogu domowego
- **who** - wypisuje liste użytkowników zalogowanych na danym systemie
- **id** - nazwa, identyfikator oraz grupa użytkownika
- **mkdir nazwa\_katalogu** - tworzy folder
- **touch nazwa\_pliku** - tworzy plik (gdy plik istnieje zmienia datę ostatniej modyfikacji)
- **rm [target]** - usuwa plik, z flagą -r usuwa katalog razem z wszystkimi podkatalogami oraz plikami
- **mv [target] [destination]** - prznosi podany plik lub katalog (-r), służy także do zmiany nazwy
- **cp [target] [destination]** - kopiuje podany plik lub katalog
- **cal** - wypisuje kalendarz
- **ps** - wypisuje informacje o procesach
- **kill PID** - po podaniu identyfikatora procesu (PID) kończy proces

- **chmod** - zmiana praw dostępu
- **echo** - wypisanie argumentów na ekran - przydatne do wypisywania zmiennych środowiskowych (o tym później)
- **cat** - połączenie oraz wypisanie zawartości przekazanych strumieni (np. pliki) na ekran
- **wc** - zlicza słowa, linie, znaki z pliku
- **tr** - zamienia frazy w plików
- **grep wzor [plik]** - wyszukuje frazę wypisując tylko linie które spełniają warunek, przyjmuje plik lub strumień tekstowy
- **less plik** - wypisanie na ekran zawartości pliku w trybie interaktywnym
- **tree [katalog]** - wypisuje strukturę katalogów
- **file plik** - wypisuje informacje o pliku

## 2.5 Kierowanie wyjściem oraz wejściem programu

Domyślnie wyjście programu jest przekierowywane na ekran, można jednak spowodować przekierowanie (*cmd* - komenda, program np. **ls**, **less**, **cat itd.**):

- *cmd > plik* - przekierowanie wyjścia do pliku, nadpisuje plik
- *cmd >> plik* - przekierowanie wyjścia do pliku, dopisuje do pliku
- *cmd 2> (2>>) plik* - przekierowanie wyjścia błędów
- *cmd &> (&>>) plik* - przekierowanie błędów i standardowego wyjścia

Aby użyć zawartości pliku jako strumienia wejściowego używamy **<**. Możliwe jest także przekierowanie wyjścia jednego programu na wejście kolejnego (jako strumień tekstowy) za pomocą symbolu **|** (pipe) - tzw. potoki. Pozwala to na łączenie funkcjonalności wielu programów w jeden.

## 3 Zadania

Zadanie 1 Założyć katalog Ala, następnie w nim katalog ma, wewnątrz katalogu ma utworzyć katalogi kota i psa.

Zadanie 2 Przejść do katalogu ./Ala/ma/kota. Założyć podobną strukturę katalogów dla Oli bez używania komendy cd. Sprawdzić strukturę katalogów

Zadanie 3 Usunąć katalog Ali wraz z podkatalogami

```

szymon@szymon-GV62-8RC:~/Folder$ grep With plik
With my them if up many. Lain week nay she them her she. Extremity so attending objection
as engrossed gentleman something.
szymon@szymon-GV62-8RC:~/Folder$ grep With < plik
With my them if up many. Lain week nay she them her she. Extremity so attending objection
as engrossed gentleman something.
szymon@szymon-GV62-8RC:~/Folder$ cat plik | grep With
With my them if up many. Lain week nay she them her she. Extremity so attending objection
as engrossed gentleman something.
szymon@szymon-GV62-8RC:~/Folder$ █

```

Rysunek 9: grep może przyjmować plik jako argument, ale może też przyjmować strumień wejściowy

Zadanie 4 Zmienić prawa dostępu do katalogu Oli tak, aby tylko użytkownik miał do niego dostęp

Zadanie 5 Przejść do najniższego katalogu, następnie z jego wnętrza wypisać pliki i katalogi z katalogu znajdującego się 2 poziomy wyżej.

Zadanie 6 Bez zmieniania katalogu zapisać strukturę katalogów z poziomu katalogu domowego do pliku w folderze Ola

Zadanie 7 Zapisz kalendarz z roku 2000 do pliku

Zadanie 8 Policz liczbę wszystkich plików i katalogów w dowolnym folderze

## 4 Rozwiązania

Zadanie 1 `mkdir -p ./Ala/ma/kota ./Ala/ma/psa`

Zadanie 2 `cd ./Ala/ma/kota;; mkdir -p ~/Ola/ma/kota ~/Ola/ma/psa; tree`

Zadanie 3 `rm -r ~/Ala/`

Zadanie 4 `chmod 700 ~/Ola/`

Zadanie 5 `cd ./Ola/ma/kota; ls ../../`

Zadanie 6 `tree ~ > ~/Ola/plik`

Zadanie 7 `cal 2000 > kalendarz2000`

Zadanie 8 `ls | wc -l` - wszystkie nieukryte pliki i katalogi, `ls -A | wc -l` - wliczając ukryte pliki i katalogi poza `./` i `../`