

Podstawy Linuksa



- System operacyjny **Linux**.
- **Powłoka** Linuksa.
- **System plików** Linuksa.
- Podstawowe **operacje** na **plikach** i **katalogach**.
- Podstawowe **operacje** na **procesach**.
- **Kompresja** i **archiwizacja**.
- **Zdalne logowanie** oraz **przesyłanie plików**.
- **Pomoce** systemowe.
- **Użyteczne polecenia/klawisze/narzędzia**.
- Używanie **Linuksa** w środowisku **MS Windows**.

System operacyjny **Linux**

- **1991: Linus Torvalds** (student Uniwersytetu Helsińskiego) zapoczątkował rozwój poprzez Internet wolnego i otwartego systemu operacyjnego **Linux** wzorowanego na systemie operacyjnym UNIX.
 - „Linux jest klonem Uniksa”
- **Linux** – wielozadaniowy i wielodostępny system operacyjny o otwartym kodzie źródłowym (ang. *open source*) ➔ najnowsza wersja jądra systemu o numerze **5.11.1** (23.02.2021).
- **Dystrybucje Linuksa**: jądro systemu + użyteczne pakiety oprogramowania, m.in. do instalacji, konfiguracji i uaktualniania systemu oraz programów użytkowych ➔ w większości darmowe.
 - Popularne: Debian, Ubuntu, Fedora, openSUSE, Gentoo, Slackware,....
- **Instalacja**:
 - **Bezpośrednio na sprzęcie** – równolegle z innym systemem operacyjnym, np. MS Windows ➔ dostępne w dystrybucjach programy instalacyjne i konfiguracyjne ułatwiają podział dysku i instalację Linuksa.
 - Poprzez **maszynę wirtualną**, np. **VirtualBox, VMware** – łatwa i wygodna instalacja, uruchamianie systemu nie wymaga restartu komputera, można mieć zainstalowanych i uruchomionych wiele systemów naraz, sprzęt pod kontrolą macierzystego systemu operacyjnego (ang. *host*).
 - **MS Windows 10: Windows Subsystem for Linux (WSL)** – umożliwia uruchamianie pewnych składowych środowiska **Linuksa** bezpośrednio w systemie **Windows** ➔ <https://docs.microsoft.com/en-us/windows/wsl/about>
 - Aby móc korzystać z aplikacji graficznego interfejsu użytkownika (GUI) Linuksa warto zainstalować **X410**: <https://x410.dev>

Powłoka **Linuksa**

- **Powłoka** (ang. *shell*) – interpreter poleceń, pośrednik między użytkownikiem a systemem operacyjnym, np. **sh**, **bash**, **csh**, **tcsh**, **zsh**.

➤ **Notacja:** \$ - symbol powłoki

\$ polecenie # objaśnienia

\$ echo \$SHELL	# wyświetla nazwę powłoki użytkownika
\$ chsh	# zmiana powłoki, np. /bin/bash
\$ ypchsh	# zmiana powłoki w SPK (Studencka # Pracownia Komputerowa)
\$ passwd	# zmiana hasła użytkownika
\$ yppasswd	# zmiana hasła użytkownika w SPK
\$ quota	# zużyte oraz przydzielone użytkownikowi # zasoby dyskowe (ilość miejsca, liczba plików)
\$ quota -s	# j.w. w formacie „ <i>human-readable</i> ”

Powłoka **Linuksa**

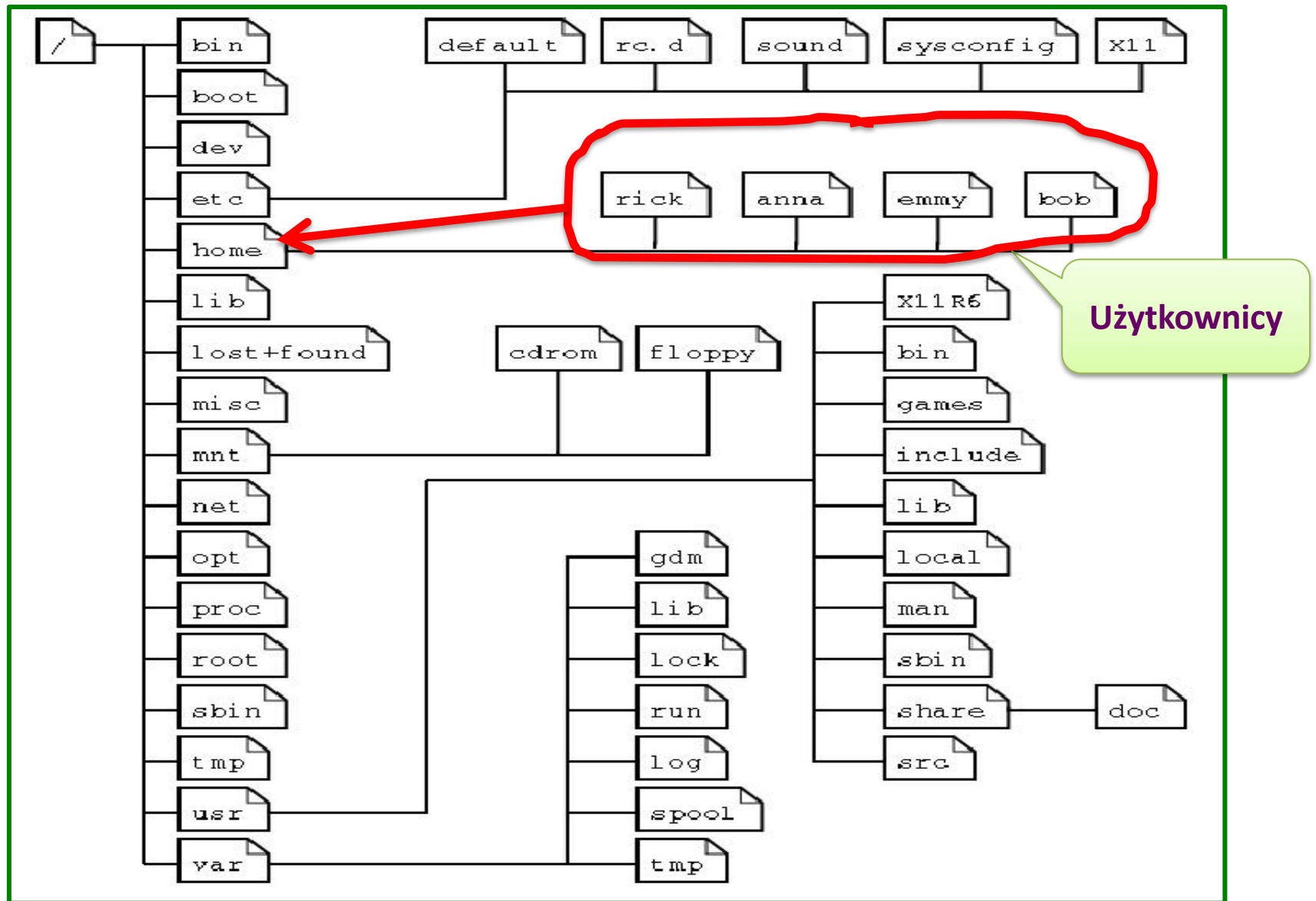
- **Powłoka** (ang. *shell*) – interpreter poleceń, pośrednik między użytkownikiem a systemem operacyjnym, np. *sh*, *bash*, *csh*, *tcsh*.

➤ **Notacja:** \$ - symbol powłoki

\$ polecenie # objaśnienia

\$ echo \$SHELL	# wyświetla nazwę powłoki użytkownika
\$ chsh	# zmiana powłoki, np. <i>/bin/bash</i>
\$ ypchsh	# zmiana powłoki w SPK (Studencka # Pracownia Komputerowa)
\$ passwd	# zmiana hasła użytkownika
\$ yppasswd	# zmiana hasła użytkownika w SPK
\$ quota	# zużyte oraz przydzielone użytkownikowi # zasoby dyskowe (ilość miejsca, liczba plików)
\$ quota -s	# j.w. w formacie „ <i>human-readable</i> ”

System plików Linuksa



Przeglądanie systemu plików

\$ ls /	# zawartość katalogu głównego („korzenia”), tzn. /
\$ tree -L 1 /	# j.w. w formie drzewa
\$ ls /home	# zawartość katalogu /home (użytkownicy)
\$ pwd	# ścieżka do bieżącego katalogu od korzenia
\$ ls	# zawartość bieżącego katalogu
\$ ls -l	# j.w. ale ze szczegółami
\$ ls -a	# pokaż też pliki ukryte (. na początku nazwy)
\$ ls -al	# połączenie powyższych opcji
\$ ls -altr	# j.w. ale w kolejności odwrotnej czasu modyfikacji

➤ **Uwaga:** **.** oznacza **bieżący** katalog,
.. oznacza katalog **macierzysty** (poziom niżej); **ls ..**
~ oznacza katalog **domowy** użytkownika; **ls ~**

Opis plików/katalogów

\$ ls -l

➤ np.

```
-rw-r--r-- 1 jan staff 2679 25 Feb 2021 14:16 plik1.txt
```

- ❑ Pierwszy znak to **rodzaj pliku**, np. - oznacza zwykły plik, d – katalog (folder), a l – dowiązanie (ang. *link*).
- ❑ 9 kolejnych znaków określa **prawa dostępu** – po 3 kolejno dla **użytkownika** (*user*), **grupy** (*group*) i **innych** (*others*):
 - r – prawo **czytania** (*read*),
 - w – prawo **pisania** (*write*),
 - x – prawo **wykonywania** (*execute*) → dla plików wykonawczych.
- ❑ 2-ga kolumna oznacza **liczbę** tzw. twardych **dowiązań**.
- ❑ 3-cia kolumna – **nazwa użytkownika** (właściciela).
- ❑ 4-ta kolumna – **nazwa grupy**, do której należy użytkownik.
- ❑ 5-ta kolumna – rozmiar pliku w bajtach.
- ❑ 4 kolejne kolumny zawierają **datę i czas** ostatniej modyfikacji.
- ❑ Ostatnia kolumna zawiera **nazwę pliku**.

Operacje na plikach/katalogach

- ❖ Nazwy plików i katalogów mogą się zawierać do **255** dowolnych znaków (małe i wielkie litery są rozróżniane) z wyjątkiem **ukośnika** (/), który służy do oddzielania nazw plików i katalogów (folderów).
 - **Uwaga:** Jeżeli nazwy zawierają spacje, to w poleceniach powłoki należy je ujmować w cudzysłów (" ").

\$ mkdir PP	# utwórz katalog o nazwie PP
\$ cd PP	# wejdź do katalogu PP
\$ pwd	# pokaż pełną ścieżkę do bieżącego katalogu
\$ cd ..	# przejdź do katalogu macierzystego
\$ cd	# przejdź do katalogu domowego , to samo: cd ~
\$ cd PP; pwd	# wróć do katalogu PP i pokaż ścieżkę
\$ touch plik0	# utwórz pusty plik o nazwie plik0
\$ ls -l	
\$ cat plik0	# wyświetl zawartość pliku plik0

Operacje na plikach/katalogach – c.d.

```
$ cat /etc/shells      # wyświetl zawartość pliku /etc/shells (nazwa ścieżkowa)
$ ls -altr ~ > plik1.txt # wyniki polecenia przekieruj (>) do pliku plik1.txt
$ ls -l
$ cat plik1.txt        # wyświetl zawartość pliku plik1.txt
$ less plik1.txt       # wyświetla plik ze stronicowaniem:
                        # spacja – strona do przodu, b – strona do tyłu, q – wyjście
$ mv plik0 pusty       # zmień nazwę pliku z plik0 na pusty
$ mkdir kat1          # utwórz podkatalog kat1
$ mv pusty kat1/      # przenieś plik pusty do podkatalogu kat1
$ ls -l kat1
$ cp plik1.txt plik2.txt # utwórz kopię pliku plik1.txt o nazwie plik2.txt
$ diff -s plik1.txt plik2.txt # porównaj pliki plik1.txt i plik2.txt
$ cp -r kat1 kat2      # utwórz kopię katalogu kat1 o nazwie kat2
$ ls -l kat2
$ rm plik2.txt         # usuń plik plik2.txt
$ rm -r kat2          # usuń katalog kat2 wraz z całą zawartością – ostrożnie!
```

Operacje na procesach

❖ **Proces** – program w trakcie wykonywania; każdy proces ma swój unikalny identyfikator w systemie: **PID**; hierarchiczna struktura procesów.

\$ ps	# wyświetla procesy bieżącej powłoki , opcje: -f, -l, -fl
\$ ps -e	# wyświetla wszystkie procesy, opcje: -efl
\$ pstree	# wyświetla całe drzewo procesów, opcja: -p dodaje PID
\$ top	# lista najważniejszych procesów z informacją o użyciu # zasobów : obciążenie procesora, pamięci itd.; q – wyjście.
\$ xterm	# uruchomienie procesu konsoli graficznej <i>xterm</i>
\$ Ctrl-Z	# zawieszenie procesu działającego w pierwszym planie
\$ bg	# uruchomienie zawieszonego procesu w tle
\$ fg	# przywrócenie procesu z tła do pierwszego planu
\$ Ctrl-C	# zakończenie procesu działającego w pierwszym planie
\$ xterm &	# uruchomienie procesu terminala <i>xterm</i> w tle
\$ ps	# ➔ odczytać identyfikator PID procesu <i>xterm</i> ➔ <i>PID_xterm</i>
\$ kill <i>PID_xterm</i>	# zakończenie procesu o PID = <i>PID_xterm</i>
\$ kill -9 <i>PID_xterm</i>	# bezwzględne zakończenie procesu („ <i>srebrna kula</i> ”)

Kompresja i archiwizacja

```
$ ls -l
```

```
$ gzip plik1.txt      # kompresuje plik plik1.txt → dodaje rozszerzenie .gz
```

```
$ ls -l
```

```
$ gunzip plik1.txt.gz  # dekompresuje plik plik1.txt.gz do pliku plik1.txt
```

```
$ ls -l
```

```
$ tar cvzf kat1.tar.gz kat1      # archiwizacja i kompresja katalogu kat1  
                                # do pliku kat1.tar.gz
```

```
$ ls -l
```

```
$ mkdir kat2; cp kat1.tar.gz kat2/; cd kat2; pwd; ls -l
```

```
$ tar xvzf kat1.tar.gz          # dekompresja i dearchiwizacja kat1.tar.gz
```

```
$ ls -l; ls -l kat1
```

```
$ cd ..; rm -r kat2; pwd; ls -l
```

```
$ touch p1.txt p2.txt p3.txt p4.txt; ls -l
```

```
$ rm *.txt      # usuń wszystkie pliki o rozszerzeniu .txt
```

```
# * zastępuje dowolny ciąg znaków, ? – jeden znak.
```

Zdalne logowanie oraz przesyłanie plików

- \$ ssh 149.156.43.64 # **logowanie** na serwer **SPK** z **zewnątrz** przez adres IP
- \$ ssh 172.30.24.11 # **logowanie** na serwer **SPK** z **pracowni** przez adres IP
- \$ ssh spk-ssh.if.uj.edu.pl # jak wyżej, tylko przez **nazwę internetową** serwera
- \$ mkdir kat2 # utworzenie katalogu **kat2** na zdalnym komputerze
- \$ exit # wylogowanie
- \$ scp kat1.tar.gz spk-ssh.if.uj.edu.pl:kat2/ # **przesłanie pliku** **kat1.tar.gz** na
zdalny komputer do katalogu **kat2**
- \$ scp -r spk-ssh.if.uj.edu.pl:kat2/ . # **przesłanie katalogu** **kat2** (dlatego opcja -r)
ze zdalnego komputera do bieżącego katalogu
- \$ ls -l
- W poleceniach **ssh** i **scp** można używać zamiennie **adresów IP** lub **nazw internetowych** odpowiednich komputerów, tzw. **hostów** (ang. *host name*).
- **Uwaga:** Jeżeli **nazwa użytkownika** na komputerze **zdalnym** jest **inna** niż na **lokalnym**, to w poleceniach **ssh** i **scp** **nazwę hosta** lub **adres IP** należy poprzedzić **nazwą użytkownika** na komputerze **zdalnym** zakończoną znakiem @, np.
- \$ ssh kowalski@spk-ssh.if.uj.edu.pl # dla użytkownika **kowalski** na serwerze SPK
- ❖ **Pomoce systemowe:** man, info, help
 - \$ man man # **opis** samego **podręcznika**, wyświetlany przez **less**
 - \$ man ls # **opis** polecenia **ls**
 - \$ info ls # bardziej **szczegółowy** (na ogół) niż w **man** opis polecenia **ls**
 - \$ help cd # opis **wbudowanych** poleceń powłoki, jak np. **cd**

Użyteczne polecenia/klawisze/narzędzia

❖ Przywoływanie poprzednich poleceń:

➤ **Wyszukiwanie** klawiszem , **zatwierdzanie** klawiszem:  [Enter]

\$ history # wyświetla **historię** poleceń wraz z ich **numerami**

\$!*numer_polecenia* # **uruchamia** polecenie o danym **numerze**

❖ Uzupełnianie wpisywanych poleceń:

➤ Klawisz tabulatora [Tab], np.

\$ ls ka[Tab] # ➔ daje: ls kat1

➤ **Uzupełnia** dowolne wpisywane w powłoce nazwy; jeżeli są **wieloznaczne**, to należy wcisnąć go **dwa** razy, **dodać** kolejny znak itd.

❖ Mysz:

➤ **lewy** klawisz wciśnięty – **zaznaczanie**, **2-krotne** kliknięcie – **zaznaczanie wyrazu**, **3-krotne** kliknięcie – **zaznaczanie całej linii**,

➤ kliknięcie **środkowego** klawisza – **wklejanie zaznaczonego wcześniej lewym klawiszem tekstu**.

❖ Edytory: emacs, gedit, kate, vi, ...

\$ emacs & # 🖱️ zapoznać się z menu

🖱️ Zapoznać się ze środowiskiem graficznym Linuksa.

Zdalne logowanie do Linuksa z MS Windows

❖ **Bitvise SSH Client** (<http://www.bitvise.com>)

- Wygodna **okienkowa** aplikacja do **zdalnego logowania** przez **ssh** oraz **zdalnego kopiowania** plików przez **sftp**:
 - W zakładce **Login** w rubryce **Host** wpisać **adres IP** zdalnego komputera, np. **149.156.43.64**, lub jego **nazwę internetową**, np. **spk-ssh.if.uj.edu.pl**, a w rubryce **Username** swoją **nazwę użytkownika** na zdalnym komputerze, np. **kowalski**.
 - W zakładce **Options** w rubryce **On login** zaznaczyć **Open Terminal** oraz **Open SFTP**.
 - W zakładce **Terminal** w rubryce **X11 Forwarding** zaznaczyć **Enable** – umożliwi to otwieranie zdalnych **aplikacji okienkowych** (np. **xterm**, **gedit**, **emacs**) lokalnie w systemie Windows (do tego potrzebny jest jeszcze np. **Xming** – patrz poniżej).
 - Nacisnąć przycisk **Login**, następnie wpisać **hasło** do swojego konta pod Linuksem ➔ pojawią się dwa okienka:
 1. Terminal **xterm** do wykonywania **poleceń** powłoki.
 2. Podwójne okno **SFTP** do **kopiowania** plików – z otwartym katalogiem **lokalnym** po **lewej** i **zdalnym** po **prawej** stronie; pliki można kopiować za pomocą przycisków **Upload** (z lokalnego na zdalny) i **Download** (ze zdalnego na lokalny).

❖ **Xming** (<http://sourceforge.net/projects/xming/>)

- **Serwer** uniksowego systemu **X Window** dla **MS Windows** – umożliwia otwieranie **okienkowych** aplikacji **linuksowych** w systemie operacyjnym **MS Windows**.
- ✓ **Uwaga:** Przed otwarciem aplikacji okienkowej Linuksa przy użyciu **Bitvise** należy uruchomić **Xming** (klikając na jego ikonę).

Zdalne logowanie do Linuksa z MS Windows

❖ PuTTY (<http://www.putty.org>)

➤ Aplikacja do zdalnego **logowania** przez **ssh**:

- W zakładce **Session** w rubryce **Host Name** wpisać **adres IP** zdalnego komputera, np. **149.156.43.64**, lub jego **nazwę internetową**, np. **spk-ssh.if.uj.edu.pl**.
- W zakładce **Connection->SSH->X11** w rubryce **X11 Forwarding** zaznaczyć **Enable X11 forwarding** – umożliwi to otwieranie zdalnych **aplikacji okienkowych** (do tego potrzebny jest jeszcze Windows X Serwer, np. **VcXsrv** – patrz poniżej).
- Nacisnąć przycisk **Open**, następnie wpisać **nazwę użytkownika** oraz **hasło** do swojego konta pod Linuksem.

❖ WinSCP (<http://winscp.net>)

➤ Aplikacja do zdalnego **kopiowania** plików przez **scp**:

- W rubryce **Nazwa_hosta** wpisać **adres IP** zdalnego komputera, np. **149.156.43.64**, lub jego **nazwę internetową**, np. **spk-ssh.if.uj.edu.pl**.
- Nacisnąć przycisk **Logowanie**, następnie wpisać **nazwę użytkownika** oraz **hasło** do swojego konta pod Linuksem → pojawi się podwójne okno (podobne do tego dla **SFTP** pod **Bitvise**) służące do **kopiowania** plików między lokalnym i zdalnym systemem.

❖ VcXsrv (<https://sourceforge.net/projects/vcxsrv/>)

- **Serwer** uniksowego systemu **X Window** dla **MS Windows** – umożliwia otwieranie **okienkowych** aplikacji **linuksowych** w systemie operacyjnym **MS Windows**., podobnie jak **Xming** (można ich używać zamiennie).
- ✓ **Uwaga:** Aby móc używać aplikacji okienkowych Linuksa, przed uruchomieniem PuTTY należy wystartować **VcXsrv** (klikając na jego ikonę).

❖ MobaXterm (<https://mobaxterm.mobatek.net>) – serwer **X Window** i klient **SSH** w jednym (prosty i wygodny w użyciu).