MATERIAŁ DLA KLASY 1 TC · POZIOM PODSTAWOWY

Linux w praktyce. Pierwsze polecenia

Przewodnik po terminalu z przykładami, wyjaśnieniami i ćwiczeniami

Po tej lekcji będziesz umieć:

sprawdzić, gdzie jesteś w systemie i wyświetlić zawartość katalogu;

poruszać się po katalogach oraz kopiować, przenosić i usuwać pliki;

odczytać tekst z pliku i wyszukać informacje;

sprawdzić podstawowe informacje o systemie i sieci;

wykonywać polecenia bezpiecznie, rozumiejąc ich skutki.

> **Najważniejsza zasada**
> Wszystkie ćwiczenia wykonuj we własnym katalogu **~/linux_lab**. Nie usuwaj nieznanych plików, nie uruchamiaj obcych skryptów i nie używaj **sudo**, jeśli nie jest potrzebne.

LINUX W PRAKTYCE / MODUŁ 01

Terminal, polecenie i ścieżka

Terminal pozwala wydawać komputerowi polecenia tekstowe. W Ubuntu i wielu innych dystrybucjach działa w nim powłoka **Bash**.

Jak wygląda polecenie?

Często składa się z nazwy, opcji oraz argumentu (np. nazwy pliku).

```
$ ls -l dokumenty
```

ls to polecenie, **-l** to opcja, a **dokumenty** to argument (nazwa katalogu). Znak **$** symbolizuje znak zachęty terminala - **nie wpisuj go**.

Co oznaczają ścieżki?

Zapis	Znaczenie
/	Katalog główny całego systemu Linux.
~	Twój katalog domowy, np. /home/uczen.
.	Bieżący katalog (tu, gdzie jesteś).
..	Katalog nadrzędny (poziom wyżej).
~/linux_lab	Katalog linux_lab w Twoim katalogu domowym.

Ważne: wielkość liter
Linux rozróżnia **Notatka.txt** i **notatka.txt**. Spacja w nazwie ma znaczenie: używaj cudzysłowów, np. **cd "Moje pliki"**.

LINUX W PRAKTYCE / MODUŁ 02

Poruszanie się po katalogach

Zanim skopiujesz lub usuniesz plik, sprawdź jego lokalizację. To podstawowy nawyk podczas pracy w terminalu.

pwd - pokaż bieżący katalog

Wypisuje pełną ścieżkę do miejsca, w którym obecnie pracujesz.

```
$ pwd
```

Przykładowy wynik: **/home/uczen**.

ls - wyświetl zawartość katalogu

Pokazuje pliki i podkatalogi. Opcje pozwalają zobaczyć więcej informacji.

```
$ ls
$ ls -l
$ ls -a
$ ls -lh
```

Opcja **-l** - szczegóły, **-a** - także pliki ukryte, **-h** - rozmiary czytelne dla człowieka (z **-l**).

cd - zmień katalog

Przechodzi do wskazanego miejsca. Bez argumentu wraca do katalogu domowego.

```
$ cd /tmp
$ cd ..
$ cd ~
$ cd -
```

cd .. - poziom wyżej; **cd ~** - do domu; **cd -** - do poprzedniego katalogu.

> **Przykład krok po kroku**
> Jesteś w **/home/uczen**. Wpisujesz **cd Pobrane**, a następnie **pwd**. Wynik powinien wskazać **/home/uczen/Pobrane**, o ile taki katalog istnieje.

LINUX W PRAKTYCE / MODUŁ 03

Tworzenie, kopiowanie i przenoszenie

Do przykładów użyjemy osobnego katalogu ćwiczeniowego. Najpierw wykonaj dwa pierwsze polecenia.

mkdir - utwórz katalog

Tworzy nowy katalog. Opcja **-p** tworzy także brakujące katalogi po drodze.

```
$ mkdir -p ~/linux_lab/dokumenty
$ cd ~/linux_lab
```

touch - utwórz pusty plik

Jeżeli plik nie istnieje, powstaje pusty plik. Dla istniejącego pliku zmienia czas modyfikacji.

```
$ touch notatka.txt
```

cp - skopiuj plik lub katalog

Kopia powstaje we wskazanym miejscu, a oryginał zostaje.

```
$ cp notatka.txt dokumenty/kopia.txt
$ cp -i notatka.txt dokumenty/kopia.txt
$ cp -r dokumenty kopia_dokumentow
```

-i pyta przed nadpisaniem, **-r** kopiuje folder z zawartością.

mv - przenieś plik albo zmień jego nazwę

Polecenie **mv** może zmienić nazwę lub przenieść plik do innego katalogu.

```
$ mv notatka.txt notatka_stara.txt
$ mv notatka_stara.txt dokumenty/
$ mv -i dokumenty/notatka_stara.txt .
```

Ostatnie polecenie przenosi plik z powrotem do obecnego katalogu. **-i** pyta przed nadpisaniem.

> **Porównanie cp i mv**
> **cp** = kopiujesz, więc masz dwa egzemplarze. **mv** = przenosisz albo zmieniasz nazwę; po przeniesieniu stara ścieżka przestaje istnieć.

LINUX W PRAKTYCE / MODUŁ 04

Czytanie, edycja i usuwanie

Najpierw zobacz, co jest w pliku, dopiero potem go edytuj lub usuwaj.

cat, less, head, tail - pokaż treść

Służą do czytania plików tekstowych w różny sposób.

```
$ cat notatka_stara.txt
$ less notatka_stara.txt
$ head -n 5 notatka_stara.txt
$ tail -n 5 notatka_stara.txt
```

cat - całość, **less** - przewijanie (wyjście: **q**), **head** - początek, **tail** - koniec.

nano - prosty edytor tekstu

Otwiera plik do edycji w terminalu.

```
$ nano notatka_stara.txt
```

Ctrl+O - zapisz, Enter - potwierdź nazwę, **Ctrl+X** - zamknij edytor.

rm i rmdir - usuwanie

Usuwa plik lub pusty katalog. Zachowaj ostrożność: zwykle nie ma kosza.

```
$ touch do_usuniecia.txt
$ rm -i do_usuniecia.txt
$ mkdir pusty
$ rmdir pusty
```

rm -i prosi o potwierdzenie. **rmdir** zadziała tylko wtedy, gdy katalog jest pusty.

> **OSTROŻNIE: rm -r i rm -rf**
> Opcja **-r** może usunąć cały katalog z zawartością. Połączenie **-rf** wymusza usuwanie bez pytań. **Nie stosuj go w ćwiczeniach** i nigdy nie kopiuj nieznanych poleceń usuwających.

LINUX W PRAKTYCE / MODUŁ 05

Szukanie plików i informacji

W pracy technika cyberbezpieczeństwa często trzeba znaleźć konkretny plik, wpis w logu albo fragment tekstu. Przed przykładami z **grep** utwórz plik **raport.txt** opisany w ramce niżej.

find - znajdź plik lub folder

Przeszukuje wskazane miejsce według podanych kryteriów.

```
$ find . -name "*.txt"
$ find ~/linux_lab -type f
```

Kropka **.** oznacza bieżący katalog. ***.txt** - pliki kończące się na .txt; **-type f** - zwykłe pliki.

grep - znajdź tekst wewnątrz pliku

Wypisuje wiersze zawierające podany tekst.

```
$ grep "BLAD" raport.txt
$ grep -n "BLAD" raport.txt
$ grep -i "blad" raport.txt
```

-n pokazuje numer wiersza, **-i** ignoruje wielkość liter. Jeśli plik nie istnieje, pojawi się komunikat błędu.

wc - policz wiersze, słowa i znaki

Przydatne do prostych analiz tekstu.

```
$ wc raport.txt
$ wc -l raport.txt
```

Opcja **-l** pokazuje liczbę wierszy.

> **Mini-próba: plik z logiem**
> Utwórz plik **raport.txt** w edytorze nano. Wpisz trzy wiersze: **OK start**, **BLAD polaczenia**, **OK koniec**. Następnie uruchom **grep -i "blad" raport.txt**. Zobaczysz drugi wiersz.

LINUX W PRAKTYCE / MODUŁ 06

Łączenie poleceń i zapis wyników

Zamiast przepisywać wyniki ręcznie, możesz przekazać je do kolejnego polecenia lub zapisać w pliku.

> - zapisz wynik do pliku

Tworzy nowy plik lub **nadpisuje** istniejący.

```
$ echo "Pierwsza notatka" > notatki.txt
$ cat notatki.txt
```

Uwaga: **>** usuwa poprzednią zawartość pliku wynikowego, jeżeli ten już istniał.

>> - dopisz do pliku

Dopisuje tekst na końcu bez usuwania wcześniejszej zawartości.

```
$ echo "Druga notatka" >> notatki.txt
$ cat notatki.txt
```

| - potok (pipe)

Przekazuje wynik jednego polecenia jako dane wejściowe następnego.

```
$ ls -l | less
$ cat notatki.txt | wc -l
```

Pierwsza komenda podaje wynik drugiej. W przykładzie **wc -l** liczy wiersze tekstu.

echo - wyświetl tekst

Wypisuje podany tekst w terminalu; z przekierowaniem zapisuje go do pliku.

```
$ echo "Witaj, Linux!"
```

> **Zapamiętaj trzy symbole**
> **>** = zapisz i nadpisz, **>>** = dopisz na końcu, **|** = przekaż wynik do następnego polecenia.

LINUX W PRAKTYCE / MODUŁ 07

Użytkownicy, uprawnienia i pomoc

Linux kontroluje dostęp do plików za pomocą właścicieli i uprawnień. W cyberbezpieczeństwie to bardzo ważne.

whoami i id - sprawdź konto

Pokażą, na jakim koncie pracujesz oraz do jakich grup ono należy.

```
$ whoami
$ id
```

ls -l - zobacz uprawnienia

Przy nazwie pliku zobaczysz m.in. właściciela i znaki **r**, **w**, **x**.

```
$ ls -l notatki.txt
```

r = odczyt, **w** = zapis, **x** = wykonywanie. Przykład: **-rw-r--r--**.

chmod - zmień uprawnienia

Może nadać prawo do wykonania pliku właścicielowi.

```
$ chmod u+x skrypt.sh
```

u = właściciel, **+** = dodaj, **x** = wykonanie. Polecenie wykonaj tylko dla własnego pliku skrypt.sh.

man i --help - sprawdź instrukcję

Gdy nie pamiętasz opcji polecenia, zobacz pomoc.

```
$ man ls
$ cp --help
```

W instrukcji **man** przewijasz treść i wychodzisz klawiszem **q**.

> **Co robi sudo?**
> **sudo** uruchamia polecenie z podwyższonymi uprawnieniami (jeżeli konto jest do tego uprawnione). Może zmienić ważne pliki systemowe. Nie używaj go bez wyraźnej potrzeby i zgody nauczyciela.

LINUX W PRAKTYCE / MODUŁ 08

System i sieć: pierwsza diagnostyka

Poniższe polecenia służą głównie do **odczytywania informacji**. Są przydatne w laboratorium cyberbezpieczeństwa.

uname, df, free - informacje o komputerze

Poznasz informacje o jądrze, wolnym miejscu na dysku i pamięci operacyjnej.

```
$ uname -a
$ df -h
$ free -h
```

ps i top - działające procesy

Pomagają sprawdzić, jakie programy pracują w systemie.

```
$ ps aux
$ top
```

Z **top** wyjdziesz klawiszem **q**.

ip - konfiguracja sieci

Pokaże interfejsy i adresy IP oraz trasę domyślną.

```
$ ip a
$ ip route
```

Wynik zależy od komputera i sieci, w której pracujesz.

ping - test łączności

Wysyła ograniczoną liczbę pakietów testowych do wskazanego adresu.

```
$ ping -c 4 1.1.1.1
```

Opcja **-c 4** kończy test po czterech próbach. Wynik zależy od zasad i dostępności sieci.

ss - sprawdź porty sieciowe

Pokazuje lokalne porty TCP/UDP nasłuchujące w systemie.

```
$ ss -tuln
```

Nie wszystkie informacje o procesach będą widoczne zwykłemu użytkownikowi.

> **Zasada pracy w laboratorium**
> Diagnostykę i testy sieciowe prowadź tylko na swoich urządzeniach lub w sieci wskazanej przez nauczyciela. Nie skanuj cudzych systemów bez zgody.

LINUX W PRAKTYCE / MODUŁ 09

Ćwiczenie prowadzone: własny katalog Linux

Przepisuj polecenia **po jednym**. Po każdej komendzie sprawdź rezultat. Nie wpisuj znaku **$**.

Krok 1. Utwórz miejsce do pracy i przejdź do niego.

```
mkdir -p ~/linux_lab/zadania
cd ~/linux_lab
pwd
ls -l
```

Krok 2. Utwórz dwa pliki i dopisz treść do pierwszego.

```
touch zadania/a.txt zadania/b.txt
echo "Linux jest systemem operacyjnym" > zadania/a.txt
echo "Uczę się terminala" >> zadania/a.txt
cat zadania/a.txt
```

Krok 3. Skopiuj plik, a następnie zmień nazwę kopii.

```
cp zadania/a.txt zadania/kopia.txt
mv zadania/kopia.txt zadania/notatka.txt
ls -l zadania
```

Krok 4. Poszukaj tekstu i sprawdź liczbę wierszy.

```
grep -n "Linux" zadania/notatka.txt
wc -l zadania/notatka.txt
find . -name "*.txt"
```

Krok 5. Usuń tylko pusty plik b.txt (z potwierdzeniem).

```
rm -i zadania/b.txt
ls -l zadania
```

> **Co powinno być na końcu?**
> W **~/linux_lab/zadania** mają znajdować się pliki **a.txt** i **notatka.txt**. Oba zawierają dwa wiersze tekstu, a **b.txt** został usunięty.

LINUX W PRAKTYCE / MODUŁ 10

Ćwiczenia samodzielne i typowe błędy

Ćwicz w katalogu **~/linux_lab**. Wykonaj zadania i pokaż nauczycielowi terminal oraz utworzone pliki.

Zadania do wykonania

- 1. Utwórz katalog **projekt**, a w nim podkatalog **raporty**.
- 2. Utwórz plik **start.txt** i wpisz do niego dwa dowolne zdania.
- 3. Skopiuj **start.txt** do **raporty/kopia.txt**. Oryginał ma pozostać.
- 4. Zmień nazwę pliku **kopia.txt** na **wynik.txt** poleceniem **mv**.
- 5. Wyświetl wszystkie pliki *.txt w katalogu **projekt** i jego podkatalogach.
- 6. Pokaż liczbę wierszy w **start.txt** oraz treść pliku **wynik.txt**.
- 7. Sprawdź swój login, adresy interfejsów sieciowych i wolne miejsce na dysku.

Jeśli terminal zgłasza błąd

Komunikat	Co sprawdzić?
`No such file or directory`	Czy plik istnieje? Czy jesteś w dobrym katalogu? Użyj pwd i ls.
`Permission denied`	Czy masz odpowiednie uprawnienia? Sprawdź ls -l; nie dodawaj automatycznie sudo.
`command not found`	Czy nie ma literówki? Czy polecenie jest zainstalowane?
`Is a directory`	Czy próbujesz potraktować katalog jak plik?

> **Dobry nawyk technika**
> Przed poleceniami **cp**, **mv** i szczególnie **rm** wykonaj **pwd** oraz **ls**. W ten sposób unikniesz pracy w niewłaściwym katalogu.

LINUX W PRAKTYCE / MODUŁ 11

Ściąga: najważniejsze polecenia

Krótka tabela do powtórzenia przed ćwiczeniami i sprawdzianem.

Polecenie	Do czego służy?
`pwd`	Pokazuje aktualny katalog.
`ls / ls -la`	Wyświetla pliki / szczegóły i ukryte elementy.
`cd / cd .. / cd ~`	Zmienia katalog / wyżej / do domu.
`mkdir / mkdir -p`	Tworzy katalog / katalogi pośrednie.
`touch`	Tworzy pusty plik lub aktualizuje znacznik czasu.
`cp / cp -r`	Kopiuje plik / cały katalog.
`mv`	Przenosi plik lub zmienia nazwę.
`rm -i / rmdir`	Usuwa plik z potwierdzeniem / pusty katalog.
`cat / less / nano`	Wyświetla / przegląda / edytuje tekst.
`head / tail`	Pokazuje początek / koniec pliku.
`grep / find`	Szuka tekstu / plików i katalogów.
`echo / wc -l`	Wypisuje tekst / liczy wiersze.
`> / >> / |`	Nadpisuje / dopisuje / przekazuje wynik.
`whoami / chmod`	Wyświetla login / zmienia uprawnienia.
`man / --help`	Otwiera instrukcję / szybką pomoc.
`ip a / ip route`	Wyświetla interfejsy IP / trasy.
`ping -c 4`	Testuje łączność czterema próbami.
`df -h / free -h`	Sprawdza miejsce na dysku / pamięć RAM.
`ps aux / top`	Wyświetla procesy.

Trzy pytania na koniec
1. Czym różni się **cp** od **mv**? **2.** Co robi **cd ..**? **3.** Dlaczego przed użyciem **rm** warto wykonać **pwd** i **ls**?