

Spis treści

Część 1. Zrozumienie terminala i powłoki

Część 2. Komendy nawigacyjne - poruszanie się w systemie plików

Część 3. Komendy do zarządzania plikami

Część 4. Wyświetlanie i przetwarzanie tekstu

Część 5. Zarządzanie uprawnieniami

Część 6. Zarządzanie użytkownikami i administracja systemu

Część 7. Informacje o systemie i monitorowanie

Część 8. Edytory tekstu – nano i vim

Część 9. Pipe, redirect i operatory

Część 10. Zarządzanie pakietami – pacman

Część 11. Konfiguracja środowiska i .bashrc

Część 12. Skrypty Bash – automatyzacja

Część 13. systemd i systemctl – usługi

Część 14. Zaawansowane techniki

Część 15. Wskazówki, bezpieczeństwo i dobre praktyki

2.4. mkdir – Make Directory

Komenda `mkdir` służy do tworzenia nowych katalogów, pojedynczych lub zagnieżdżonych, z opcjonalnym ustawieniem uprawnień.

```
$ mkdir nowy_folder  
$ mkdir -p a/b/c  
$ mkdir -m 755 folder
```



Opcja `-p` tworzy całą strukturę zagnieżdżonych katalogów, natomiast `-m` pozwala od razu nadać konkretne uprawnienia (np. `755`).

Praktyczny przykład użycia – katalog na projekty

```
$ mkdir -p ~/projekty/{python,bash,java}
```



Powyższa komenda utworzy w katalogu `~/projekty` trzy podkatalogi: `python`, `bash` i `java`, w wygodnej strukturze do organizowania swoich projektów.

3.2. cp – kopiowanie plików i katalogów

Komenda `cp` kopiuje pliki lub całe katalogi do podanej lokalizacji, zachowując oryginał w miejscu źródłowym.

```
$ cp plik.txt backup_plik.txt
$ cp plik.txt /tmp/plik.txt
$ cp -r katalog1 katalog_kopia
$ cp -v plik.txt /tmp
$ cp -i plik.txt /tmp/plik.txt
```



Opcja `-r` usuwa katalog wraz z całą zawartością, `-i` pyta o potwierdzenie, `-f` wymusza usunięcie, a `-v` pokazuje każdy usuwany element.

Backup katalogu Dokumenty

By wykonać prostą kopię zapasową całego katalogu `Dokumenty` do katalogu `Dokumenty_backup` w katalogu domowym, możesz użyć:

```
$ cp -r ~/Dokumenty ~/Dokumenty_backup
```



Taka kopia przydaje się przed większymi porządkami lub zmianami w strukturze plików i katalogów.

3.4. rm – trwałe usuwanie plików i katalogów

Komenda `rm` usuwa pliki i katalogi trwale, bez przenoszenia do kosza – odzyskanie danych po jej użyciu jest najczęściej trudne lub niemożliwe.

```
$ rm plik.txt
$ rm -r katalog
$ rm -i plik.txt
$ rm -f plik.txt
$ rm -v plik.txt
```



Opcja `-r` usuwa katalog wraz z całą zawartością, `-i` pyta o potwierdzenie, `-f` wymusza usunięcie, a `-v` pokazuje każdy usuwany element.

Uwaga: `rm` jest bezlitosne

W Linuksie komenda `rm` nie korzysta z kosza graficznego – po usunięciu pliku standardowymi narzędziami nie istnieje prosta opcja „cofnij”.

Szczególnie ostrożnie używaj kombinacji `rm -rf`, która bez pytania usuwa katalog wraz z całą zawartością, nawet jeśli jest bardzo rozbudowana.

Praktyczne przykłady

```
$ rm *.tmp
$ rm -rf /moj_katalog
```



Pierwsza komenda usuwa wszystkie pliki z rozszerzeniem `.tmp` w bieżącym katalogu, natomiast druga usuwa katalog `/moj_katalog` razem z całą zawartością. Z tego polecenia korzystaj tylko, jeśli masz pewność, że nie ma tam ważnych danych.

4.2. less / more – przeglądanie długich plików

Komendy `less` i `more` pozwalają wygodnie przeglądać długie pliki strona po stronie, zamiast wyświetlać wszystko naraz.

```
$ less plik.txt  
$ more plik.txt
```



`less` jest bardziej rozbudowane i wygodne – umożliwia przewijanie w górę i w dół, wyszukiwanie tekstu, szybki skok na początek lub koniec pliku.

```
$ less /var/log/syslog  
$ journalctl | less
```



W ten sposób możesz wygodnie analizować logi systemowe lub inne dłuższe pliki tekstowe bez zalewania terminala dziesiątkami ekranów tekstu.

Najważniejsze skróty w less

Klawisz	Działanie
Space	Następna strona
b	Poprzednia strona
/tekst	Wyszukaj tekst w dół pliku
n	Następne dopasowanie wyszukiwania
g	Początek pliku
G	Koniec pliku
q	Wyjście z programu

7.3. free – pamięć RAM i swap

Komenda `free` pokazuje ilość dostępnej i używanej pamięci RAM oraz przestrzeni wymiany (swap), co przydaje się przy diagnozowaniu spowolnień systemu.

```
$ free -h
```



Opcja `-h` ponownie zapewnia wygodny format jednostek, a kolumny `used`, `free`, `available` pomagają ocenić, czy system zaczyna korzystać intensywnie ze swapu.

Interpretacja „available” vs „free”

W nowoczesnych systemach Linux duża część pamięci jest używana jako cache, dlatego kolumna `free` może być niska, mimo że system nie jest faktycznie przeciążony.

Bardziej wiarygodnym wskaźnikiem jest `available`, które pokazuje pamięć, jaką system może przydzielić nowym procesom bez agresywnego czyszczenia cache.