System plików, ścieżki, pwd, cd, ls

Poziom: podstawy **Dotyczy: wszystkie dystrybucje** **Czas: ~12 min**

W Linuksie nie ma dysku `C:` — jest jedno drzewo od `/`, a wszystkie nośniki są w nie wpięte. Ta strona uczy się w nim poruszać i czytać wynik `ls -l`.

Skróty użyte na tej stronie (3)

Skrót	Co znaczy
RHEL	Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa
SELinux	Security-Enhanced Linux: dodatkowa warstwa uprawnień w jądrze RHEL
SLES	SUSE Linux Enterprise Server: płatna dystrybucja Linuksa firmy SUSE, konkurent Rocky/RHEL

Fiszka

TYLKO ODCZYT

Sprawdzenie ścieżki logicznej

```
pwd
```

Wypisujemy ścieżkę zapamiętaną przez powłokę w zmiennej `$PWD` — tę, którą przyszliśmy, niekoniecznie fizyczną.

Przykład:

```
/var/www
```

pwd = print working directory: pokaż, gdzie jestem

TYLKO ODCZYT

Ścieżka fizyczna, po rozwinięciu dowiązań

```
pwd -P
```

Rozwijamy każde dowiązanie symboliczne po drodze i pokazujemy rzeczywiste położenie w drzewie. Różni się od zwykłego `pwd` tylko wtedy, gdy dotarliśmy tu przez dowiązanie — tu `/var/www` jest dowiązaniem do `/srv/www`.

Przykład:

```
/srv/www
```

pwd = print working directory: pokaż, gdzie jestem

TYLKO ODCZYT

Przejście do katalogu — ścieżka bezwzględna

```
cd /etc
```

Zaczyna się od `/`, więc zadziała identycznie niezależnie od tego, gdzie wcześniej byliśmy. Powodzenie jest ciche — brak komunikatu.

cd = change directory: przejdź do katalogu

TYLKO ODCZYT

Przejście względne

```
cd ../log
```

Liczy od katalogu bieżącego: krok wyżej, potem w dół do `log`. Zadziała tylko wtedy, gdy taki katalog rzeczywiście tam jest.

cd = change directory: przejdź do katalogu

TYLKO ODCZYT

Powrót do katalogu domowego

```
cd
```

Bez argumentu przenosi do `$HOME` — to samo co `cd ~`.

cd = change directory: przejdź do katalogu

TYLKO ODCZYT

Katalog domowy innego użytkownika

```
cd ~adm_user
```

Tylda z dopisaną nazwą rozwija się do katalogu domowego tego użytkownika, a nie naszego.

cd = change directory: przejdź do katalogu

TYLKO ODCZYT

Powrót do poprzedniego katalogu

```
cd -
```

Jedyna odmiana `cd`, która coś wypisuje: przełączamy do katalogu zapamiętanego w `$OLDPWD` i od razu widzimy, dokąd trafiliśmy.

Przykład:

```
/var/log
```

cd = change directory: przejdź do katalogu

TYLKO ODCZYT

Zawartość katalogu

```
ls
```

Domyślny widok: nazwy plików i katalogów bez tych zaczynających się od kropki.

Przykład:

```
Desktop  Documents  Downloads  Pictures  projekt  skrypty.sh
```

ls = list: wypisz zawartość katalogu

TYLKO ODCZYT

Pliki ukryte

```
ls -a
```

Dokładamy wpisy zaczynające się od kropki, łącznie z `.` (ten katalog) i `..` (nadrzędny). `-A` pokazuje ukryte bez tych dwóch.

Przykład:

```
.  ..  .bash_history  .bashrc  .config  .ssh  Desktop  Documents  Downloads  Pictures
projekt  skrypty.sh
```

ls = list: wypisz zawartość katalogu

TYLKO ODCZYT

Długi format z czytelnymi rozmiarami

```
ls -lh
```

Prawa, właściciel, rozmiar i data w jednej linii na wpis; `-h` zamienia bajty na `4.0K` / `12M` / `1.3G`.

Przykład:

```
total 24K
drwxr-xr-x. 2 adm_user adm_user 4.0K Sep 10 14:02 Desktop
drwxr-xr-x. 2 adm_user adm_user 4.0K Sep 10 14:02 Documents
-rw-r--r--. 1 adm_user adm_user  12K Sep 16 19:41 skrypty.sh
```

ls = list: wypisz zawartość katalogu

TYLKO ODCZYT

Sortowanie po dacie

```
ls -ltr
```

Najnowsze pliki lądują na dole listy, tuż nad znakiem zachęty — wygodne, gdy katalog jest długi i szukamy tego, co się właśnie zmieniło.

Przykład:

```
drwxr-xr-x. 2 adm_user adm_user 4096 Sep 10 14:02 Desktop
drwxr-xr-x. 2 adm_user adm_user 4096 Sep 10 14:02 Documents
-rw-r--r--. 1 adm_user adm_user 12043 Sep 16 19:41 skrypty.sh
```

ls = list: wypisz zawartość katalogu

TYLKO ODCZYT

Sortowanie po rozmiarze

```
ls -lS
```

Największy plik na górze. Odwrócenie kolejności to dodatkowe `-r`, tak jak przy sortowaniu po dacie.

Przykład:

```
-rw-r--r--. 1 adm_user adm_user 12043 Sep 16 19:41 skrypty.sh
drwxr-xr-x. 2 adm_user adm_user  4096 Sep 10 14:02 Desktop
drwxr-xr-x. 2 adm_user adm_user  4096 Sep 10 14:02 Documents
```

ls = list: wypisz zawartość katalogu

TYLKO ODCZYT

Sam katalog, nie jego zawartość

```
ls -ld /var/log
```

Bez `-d` `ls` wchodzi do środka katalogu i pokazuje, co w nim jest. Z `-d` traktuje sam katalog jako pojedynczy wpis — jedyny sposób, żeby zobaczyć jego własne prawa.

Przykład:

```
drwxr-xr-x. 12 root root 4096 Sep 17 09:41 /var/log
```

ls = list: wypisz zawartość katalogu

TYLKO ODCZYT

Numery inode w wynikach

```
ls -li
```

Dokładamy pierwszą kolumnę z numerem inode. Dwa wpisy z tym samym numerem to dwie nazwy tego samego pliku — dowiązanie twarde.

Przykład:

```
1704892 -rw-r--r--. 1 adm_user adm_user 12043 Sep 16 19:41 skrypty.sh
1704811 drwxr-xr-x. 2 adm_user adm_user  4096 Sep 10 14:02 Desktop
```

ls = list: wypisz zawartość katalogu

TYLKO ODCZYT

Metadane pliku

```
stat /etc/hostname
```

Pełny zestaw metadanych: numer inode, prawa, właściciel, rozmiar i trzy znaczniki czasu — czegoś więcej niż pokazuje `ls -l`.

Przykład:

```
  File: /etc/hostname
  Size: 12          Blocks: 8       IO Block: 4096   regular file
Device: fd00h/64768d  Inode: 34567890    Links: 1
Access: (0644/-rw-r--r--)  Uid: (0/root)   Gid: (0/root)
Modify: 2026-08-14 11:02:44.000000000 +0200
Change: 2026-08-14 11:02:44.000000000 +0200
```

TYLKO ODCZYT

Drzewo katalogów

```
tree -L 2 /etc
```

Rysujemy strukturę katalogów graficznie, do podanej głębokości — tu dwa poziomy. Nie jest instalowane domyślnie.

Przykład:

```
/etc
├── NetworkManager
│   ├── conf.d
│   └── system-connections
├── cron.d
├── fstab
├── hostname
├── hosts
├── profile.d
│   ├── lang.sh
│   └── vim.sh
└── ssh
    ├── ssh_config
    └── sshd_config

6 directories, 6 files
```

□□ TYLKO ODCZYT

Wszystkie flagi

Flaga `ls`	Co robi	Kiedy używasz
`-l`	długi format z prawami i datą	zawsze, gdy liczy się coś poza nazwą
`-a`	pokazuje pliki ukryte	„katalog jest pusty", a jednak coś w nim jest
`-h`	rozmiary czytelne dla człowieka	razem z `-l`
`-t` / `-S`	sortowanie po czasie / rozmiarze	szukasz najnowszego albo największego

Flaga `ls`	Co robi	Kiedy używasz
`-r`	odwraca sortowanie	`-ltr` w długim katalogu
`-d`	katalog jako wpis, nie jako zawartość	sprawdzasz prawa samego katalogu
`-i`	numer inode	podejrzewasz dowiązania twarde

Uważaj

- `ls -l katalog` pokaże **zawartość** katalogu; sam katalog pokazuje `ls -ld katalog`.
- `cd ..` cofa po ścieżce **logicznej**, więc z katalogu będącego dowiązaniem wracasz gdzie indziej niż na dysku (`cd -P ..` idzie fizycznie).
- `/tmp` jest sprzątane automatycznie (`systemd-tmpfiles`, domyślnie po 10 dniach) — nic trwałego tam nie trzymaj.

Z życia wzięte

□□ Z ŻYCIA WZIĘTE

Piątek, 8:40. Skrypt kopii na `bkp01` robi `cd /var/www` (dowiązanie do `/srv/www`), potem `cd ../archiwum`, a plik ląduje w `/var/archiwum` zamiast w `/srv/archiwum`. Tomek godzinę szuka zaginionej kopii, zanim sprawdza `pwd -P` i widzi, że `cd ..` cofa po ścieżce logicznej, nie fizycznej. Poprawia skrypt na ścieżki bezwzględne, żeby pomyłka się nie powtórzyła.

Po co to istnieje

Pojęcie: system plików. Struktura, w jakiej dane są zorganizowane na nośniku: katalogi, pliki, ich metadane i sposób adresowania bloków (przykłady: `ext4`, `xfs`, `btrfs`). To nie to samo, co „drzewo katalogów" — drzewo to widok, który Linux Ci pokazuje, a system plików to konkretna struktura pod spodem, osobna dla każdego nośnika.

Na Windowsie każdy dysk ma literę i własny korzeń. W Linuksie korzeń jest jeden — `/` — a każdy kolejny system plików zostaje **wpięty** (zamontowany) w jakiś katalog tego jednego drzewa. Dodanie drugiego dysku nie tworzy nowej litery, tylko katalog, w którym ten dysk „wychodzi na powierzchnię", na przykład `/var/lib/dane`. Konsekwencja jest praktyczna: ścieżka nie mówi, na

którym dysku leży plik — `/var/log/messages` i `/home/adm_user/notatki.txt` mogą być na dwóch różnych macierzach, a wyglądają jak sąsiedzi. Od tego jest `df -h /sciezka`.

Analogia

Drzewo katalogów to **biurowiec z piętrami i numerami pokoi**.

Ścieżka bezwzględna to pełny adres: „ulica Przykładowa 5, piętro 3, pokój 312". Zaczyna się od budynku, więc zadziała niezależnie od tego, gdzie stoisz — dlatego w skryptach, jednostkach systemd i crontabach pisze się tylko takie ścieżki. Ścieżka względna to wskazówka od miejsca, w którym jesteś: „dwa pokoje dalej" — krótsza, ale bez punktu wyjścia bezużyteczna. `..` to schody w górę, `.` to pokój, w którym stoisz, `~` to Twoje biurko, a `cd -` to „wróć tam, gdzie byłem przed chwilą".

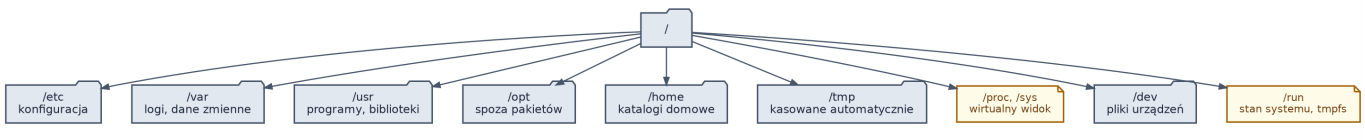
Analogia niesie ograniczenie, które tłumaczy realną pułapkę. W budynku bywają **przejścia skrótowe** — drzwi z pokoju 312 prosto do magazynu w przybudówce. Wchodzisz nimi i dalej czujesz się na trzecim piętrze budynku głównego, bo tak wszedłeś. Ale „schodami w górę" wylądujesz w przybudówce, a nie tam, gdzie się spodziewasz. Tym przejściem jest **dowiązanie symboliczne**: `pwd` podaje adres drogą, którą przyszedłeś, a nie rzeczywistą — od tej drugiej jest `pwd -P`.

Jak to działa

Rozwiń: jak to działa pod spodem

Najważniejsze katalogi (FHS)

Układ katalogów opisuje **Filesystem Hierarchy Standard**. To nie konwencja z przyzwyczajenia — pakiety, SELinux i systemd zakładają go wprost.



Najważniejsze katalogi drzewa Linuksa, wpięte pod jednym korzeniem / (kliknięcie otwiera pełny rozmiar)

Katalog	Co tam jest	Uwagi
`/etc`	konfiguracja systemu	tekstowa, nigdy binaria
`/var`	dane zmienne: logi, kolejki, bazy, cache	`/var/log`, `/var/lib`, `/var/spool`; rośnie
`/usr`	programy i biblioteki z pakietów	`/usr/bin`, `/usr/lib`, `/usr/share`
`/opt`	oprogramowanie spoza pakietów dystrybucji	podkatalog na produkt, np. `/opt/agent`
`/home`	katalogi domowe użytkowników	często osobny wolumen lub NFS
`/tmp`	pliki tymczasowe, kasowane automatycznie	każdy pisze (sticky bit); na dłużej `/var/tmp`
`/proc`	wirtualny widok na procesy i jądro	pliki mają rozmiar 0, a coś zawierają
`/sys`	wirtualny widok na urządzenia i sterowniki	tędy zmienia się parametry sprzętu
`/dev`	pliki urządzeń	tworzone przez `udev`
`/run`	stan działającego systemu: PID-y, gniazda	tmpfs, znika po restarcie

Do kompletu: `/boot` (jądro i initramfs), `/root`, `/srv` (dane serwowane przez usługi), `/mnt` i `/media` (punkty montowania). Na RHEL 7+ i Debianie 10+ obowiązuje *usrmerge*: `/bin`, `/sbin` i `/lib` są dowiązaniami do `/usr`, więc `/bin/ls` i `/usr/bin/ls` to ten sam plik.

Ścieżki i skróty

> **Pojęcie: katalog roboczy.** Katalog, względem którego powłoka rozwiązuje każdą ścieżkę niezaczynającą się od `/`. Każdy proces ma swój własny katalog roboczy, odziedziczony po procesie, który go uruchomił, i zmieniany wyłącznie przez `cd` w bieżącej powłoce — dlatego `cd` uruchomione w podprocesie albo w skrypcie nie zmienia katalogu roboczego powłoki, z której ten skrypt wywołano. `pwd` po prostu pokazuje jego bieżącą wartość.

Zapis ścieżki ma kilka wariantów, wszystkie liczone względem katalogu roboczego oprócz pierwszego:

Zapis	Znaczenie
`/etc/ssh/sshd_config`	bezwzględna — od korzenia, zawsze ta sama
`ssh/sshd_config`	względna — od katalogu bieżącego

Zapis	Znaczenie
`.` / `..`	katalog bieżący (stąd `./skrypt.sh`) / nadrzędny
`~`	katalog domowy bieżącego użytkownika (`~adm_user` — cudzy)
`-`	poprzedni katalog, tylko w `cd` (z `$OLDPWD`)

`cd` bez argumentu przenosi do `$HOME` — to nie błąd składni, tylko skrót. `cd -` skacze do poprzedniego katalogu, wypisując dokąd; powtórzone przełącza tam i z powrotem, co jest wygodne przy pracy w dwóch odległych miejscach drzewa.

`pwd` kontra `pwd -P`

Bash pamięta w `PWD` ścieżkę, którą przyszedłeś, i domyślnie ją pokazuje. Jeśli po drodze było dowiązanie symboliczne, ścieżka logiczna i fizyczna się rozjeżdżają — dokładnie to widziałeś w Fiszce: `/var/www` (dowiązanie) i `/srv/www` (cel) dają różną odpowiedź `pwd` i `pwd -P`. Ciąg dalszy tego samego przykładu:

```
cd /var/www
cd ..
pwd
```

```
/var
```

Zaskakuje: `cd ..` cofnęło po ścieżce logicznej, więc trafiłeś do `/var`, a nie do `/srv`. Fizycznie zachowuje się `cd -P ..`. To samo dotyczy programów — `/bin/pwd` z coreutils domyślnie pokazuje ścieżkę fizyczną, a `pwd` wbudowane w bash logiczną.

Czytanie wyniku `ls -l`

```
ls -lh /etc/hosts
```

```
-rw-r--r--. 1 root root 213 Aug 14 11:02 /etc/hosts
```

Kolumny po kolei: typ (`-` plik, `d` katalog, `l` dowiązanie, `c`/`b` urządzenie); prawa w trzech trójkach (właściciel, grupa, pozostali); kropka oznaczająca kontekst SELinux (`+`, gdy są ACL-e); liczba dowiązań twardych; właściciel; grupa; rozmiar; czas modyfikacji; nazwa.

Dla katalogu liczba dowiązań to nie „ile plików w środku", tylko liczba wpisów wskazujących na ten katalog: jeden z rodzica, jeden z `.` i po jednym z `..` każdego podkatalogu. Rozmiar katalogu to rozmiar listy nazw, zwykle 4096 bajtów — sumę zawartości liczy `du -sh`.

Pliki ukryte, inode, `stat`

Plik „ukryty" to plik, którego nazwa zaczyna się od kropki — nie ma tu żadnego atrybutu, `ls` po prostu pomija takie nazwy bez `-a`. Wzięło się to z przypadku: wczesne `ls` miało odfiltrować wpisy `.` i `..`, a zrobiono to najprościej — odrzucając wszystko z kropką na początku. Konfiguracja użytkownika (`.bashrc`, `.ssh/`) korzysta z tego do dziś.

> **Pojęcie: inode.** Numer wpisu w tablicy systemu plików, pod którym leżą metadane pliku i adresy jego bloków na dysku — wszystko poza samą nazwą. Nazwa w katalogu jest tylko etykietą wskazującą na inode, dlatego dwie różne nazwy, nawet w różnych katalogach tego samego systemu plików, mogą wskazywać ten sam inode i tę samą zawartość. To właśnie jest dowiązanie twarde. Numer inode pokazuje `ls -li`, a pełne metadane pod nim — `stat`, z wynikiem takim jak w Fiszce.

`Modify` w wyniku `stat` to zmiana treści, `Change` — zmiana metadanych (prawa, właściciel); `ls -l` pokazuje `Modify`.

Krok po kroku

1. Ustal punkt wyjścia i wróć do domu.

```
pwd; cd /var/log; pwd; cd; pwd; cd -; pwd
```

Ostatnia komenda wraca do `/var/log` i sama wypisuje tę ścieżkę.

2. Obejrzyj wpis katalogu zamiast jego zawartości.

```
ls -ld /var/log
```

Wynik jak w Fiszce. Bez `-d` dostałbyś listę kilkudziesięciu logów.

3. Znajdź najnowsze i największe pliki.

```
ls -ltrh /var/log | tail -n 5
ls -lSh /var/log | head -n 5
```

`-ltr` układa najnowsze na dole, tuż nad znakiem zachęty — wygodne w długim katalogu.

4. Zobacz strukturę i szczegóły: `tree -L 2 -d /etc` oraz `stat /etc/hostname`.

`tree` nie jest instalowane domyślnie: `dnf install tree` (RHEL/Rocky), `apt install tree` (Debian/Ubuntu), `zypper install tree` (SLES).

Częste błędy

`ls -l katalog` zamiast `ls -ld katalog`.

Chcesz sprawdzić prawa katalogu `/srv/dane`, a dostajesz:

```
total 8
-rw-r--r--. 1 root root 42 Sep 17 09:12 plik1.txt
-rw-r--r--. 1 root root 17 Sep 17 09:12 plik2.txt
```

To prawa plików w środku, nie katalogu. Wniosek „prawa są dobre" jest więc o czymś innym, niż myślisz — a to katalog decyduje, czy ktoś w ogóle wejdzie do środka. Poprawnie: `ls -ld /srv/dane`.

`cd` na plik albo na literówkę.

```
cd /etc/hosts
cd /var/logs
```

```
bash: cd: /etc/hosts: Not a directory
bash: cd: /var/logs: No such file or directory
```

Pierwszy komunikat mówi „ścieżka istnieje, ale to nie katalog", drugi — „nie ma takiej ścieżki". Trzeci wariant, `Permission denied`, oznacza brak prawa `x` na katalogu, które odpowiada właśnie za wchodzenie do środka.

`cd ..` po wejściu przez dowiązanie symboliczne.

Skrypt robi `cd /var/www` (dowiązanie do `/srv/www`), potem `cd ../backup` i zapisuje tam kopię. Kopia ląduje w `/var/backup`, nie w `/srv/backup`, bo bash cofa po ścieżce logicznej — a `pwd` pokazuje dokładnie to, co skrypt „miał na myśli", więc szukanie przyczyny trwa. W skryptach używaj ścieżek bezwzględnych albo `cd -P`.

Kopiowanie, które gubi pliki ukryte.

```
cp -r /home/adm_user/projekt/* /var/tmp/kopia/
```

Gwiazdka jest rozwijana przez powłokę, a ta domyślnie nie dopasowuje nazw z kropką na początku. `.env`, `.git/` i `.config/` zostają w miejscu, a kopia wygląda na kompletną. Rozwiązania: skopiuj katalog zamiast zawartości (`cp -r /home/adm_user/projekt /var/tmp/kopia`), włącz `shopt -s dotglob` albo użyj `rsync -a zrodlo/ cel/`.

□□ Ćwiczenia

> Wszystko poniżej wykonujesz na własnej maszynie, w katalogu `/tmp`. Nic nie wymaga uprawnień administratora.

Ćwiczenie 1 — drzewo z dowiązaniem

Zbuduj małe drzewo i sprawdź na nim zachowanie ścieżek.

```
cd /tmp
mkdir -p cwiczenie/{alfa,beta/gamma}
touch cwiczenie/alfa/plik.txt cwiczenie/.ukryty
ln -s /tmp/cwiczenie/beta/gamma /tmp/cwiczenie/skrot

cd /tmp/cwiczenie
ls
ls -a
ls -ld skrot
cd skrot
pwd
pwd -P
cd ..
pwd
```

Odpowiedź

`ls` pokaże `alfa beta skrot`, a `ls -a` dołoży `.`, `..` i `.ukryty`. `ls -ld skrot` da wiersz zaczynający się od `l` ze strzałką do `/tmp/cwiczenie/beta/gamma`.

Po `cd skrot`: `pwd` zwróci `/tmp/cwiczenie/skrot`, a `pwd -P` — `/tmp/cwiczenie/beta/gamma`. Po `cd ..` wylądujesz w `/tmp/cwiczenie`, mimo że fizycznie katalogiem nadrzędnym jest `/tmp/cwiczenie/beta`. Sprawdź różnicę: `cd skrot && cd -P .. && pwd`. Sprzątanie: `cd /tmp && rm`

`-rf cwiczenie`.

Pytania kontrolne

1. `ls -l /srv/dane` wypisało listę plików, a Ty chciałeś zobaczyć prawa samego katalogu. Dlaczego tak się stało?

Odpowiedź

Bo `ls` dostał katalog i domyślnie wypisuje jego zawartość. Flaga `-d` mówi „potraktuj argument jak zwykły wpis i nie zaglądaj do środka", więc właściwa komenda to `ls -ld /srv/dane`. Rozróżnienie ma znaczenie, bo o wejściu do katalogu decydują jego własne prawa (`x`), a nie prawa plików w nim leżących.

2. `/var/www` jest dowiązaniem do `/srv/www`. Wchodzisz `cd /var/www`, robisz `cd ..` i `pwd` pokazuje `/var`. Dlaczego nie `/srv`?

Odpowiedź

Bo `cd` w bashu operuje na ścieżce logicznej z `$PWD`: skleja ją z `..` tekstowo i dopiero wynik sprawdza w systemie plików. Fizycznie katalogiem nadrzędnym jest `/srv`, ale bash o to nie pyta. Prawdę pokaże `pwd -P`, a fizycznie cofa `cd -P ..`. W skryptach bezpieczniej użyć ścieżki bezwzględnej niż liczyć na `..`.

3. Kopiujesz projekt przez `cp -r projekt/* kopia/` i w kopii brakuje `.env` oraz `.git/`. Gdzie jest przyczyna?

Odpowiedź

W powłoce, nie w `cp`. To powłoka rozwija `*` w listę nazw i domyślnie pomija te z kropką na początku — `cp` dostaje listę bez plików ukrytych i wykonuje polecenie dokładnie tak, jak je otrzymał. Najprostsza naprawa: skopiuj katalog zamiast jego zawartości (`cp -r projekt kopia`); alternatywy to `shopt -s dotglob` albo `rsync -a projekt/ kopia/`.

4. Dodajesz do serwera drugi dysk. Dlaczego nie pojawi się jako `D:` i skąd wiadomo, gdzie trafiły dane?

Odpowiedź

Bo jest jedno drzewo od `/`, a nośniki są w nie montowane: nowy dysk staje się widoczny w wybranym katalogu, na przykład `/var/lib/dane`, i od tej chwili ścieżka nie mówi nic o tym, na czym fizycznie leży plik. Które urządzenie obsługuje daną ścieżkę, powie `df -h /var/lib/dane`, `findmnt /var/lib/dane` albo pole `Device` w `stat`.

Powiązane: Terminal, TTY, konsola, powłoka · Anatomia komendy · Prawa dostępu: rwx, chmod, chown · Dowiązania twarde i symboliczne, inode · Montowanie i df

Źródła: `man 7 file-hierarchy` · `man 1 ls` · `man 1 pwd` · `help cd` · `man 1 stat` · FHS 3.0

Revision #19
Created 2026-09-17 11:56:11 UTC by Tomasz Konieczny
Updated 2026-10-03 19:07:22 UTC by Tomasz Konieczny