Linux a Windows: te same zadania, dwa światy

Poziom: podstawy **Dotyczy: Windows Server 2022, Rocky Linux 9** **Czas: ~25 min**

Ta strona pokazuje te same codzienne czynności administratora wykonane najpierw w PowerShellu, potem w bashu. Nie uczymy się tu nowego systemu od zera, tylko przekładamy to, co już umiemy z Windows, na słownictwo Linuksa. Wracamy tu za każdym razem, gdy szukamy linuksowego odpowiednika znanej komendy Windows.

Skróty użyte na tej stronie (5)

Skrót	Co znaczy
CMD	skrót od polecenia command, dawna nazwa wiersza poleceń Windows sprzed PowerShella
EL	Enterprise Linux: rodzina RHEL i jej klonów (Rocky, Alma, Oracle Linux)
IP	Internet Protocol: adres komputera w sieci; w elektryce: stopień ochrony obudowy (Ingress Protection)
PID	Process ID: numer procesu w systemie
SSH	Secure Shell: szyfrowane zdalne logowanie do serwera

Fiszka

TYLKO ODCZYT

Sprawdzenie folderu, w którym jesteśmy, i listy plików: Get-ChildItem oraz ls

```
Get-Location
Get-ChildItem
```

```
pwd
ls -la
```

Obie komendy najpierw pokazują pełną ścieżkę bieżącego folderu, a potem listę plików i podfolderów w nim. Flaga `-la` w bashu pokazuje też pliki ukryte i dodatkowe informacje o prawach dostępu.

Przykład:

```
PS C:\Users\tomek> Get-Location


Path
----
C:\Users\tomek


[tomek@docker01 ~]$ pwd
/home/tomek
[tomek@docker01 ~]$ ls -la
-rw-r--r-- 1 tomek tomek 512 27 wrz 10:15 notatki.txt
```

□□ **pwd** = print working directory: pokaż, gdzie jestem · **ls** = list: wypisz zawartość katalogu

□□ **TYLKO ODCZYT**

Wyświetlenie zawartości pliku: Get-Content oraz cat

```
Get-Content C:\Users\tomek\notatki.txt
```

```
cat /home/tomek/notatki.txt
```

Obie komendy wypisują cały plik tekstowy na ekran, od pierwszej do ostatniej linii. Dla bardzo długich plików na Linuksie wygodniejszy bywa `less`, który pozwala przewijać tekst zamiast wypisywać go od razu w całości.

Przykład:

```
PS C:\Users\tomek> Get-Content C:\Users\tomek\notatki.txt
Backup dc02 zaplanowany na 22:00


[tomek@docker01 ~]$ cat /home/tomek/notatki.txt
```

```
kontener gitea restartowany 26.09, dziala poprawnie
```

cat = concatenate: sklej i wypisz pliki

TYLKO ODCZYT

Szukanie tekstu w pliku: Select-String oraz grep

```
Select-String -Path C:\logs\app.log -Pattern "blad"
```

```
grep "blad" /var/log/app.log
```

Obie komendy przeszukują wskazany plik linia po linii i wypisują tylko te linie, w których pojawia się szukany tekst. Dopasowanie działa też na kilku plikach naraz, wystarczy podać więcej ścieżek albo wzorzec z gwiazdką.

Przykład:

```
PS C:\> Select-String -Path C:\logs\app.log -Pattern "blad"

app.log:214:14:32:05 blad polaczenia z baza CDN_Zielona

[tomek@docker01 ~]$ grep "blad" /var/log/app.log
14:32:05 blad polaczenia z kontenerem gitea
```

grep = global regular expression print: szukaj wzorca w tekście

TYLKO ODCZYT

Sprawdzenie działających procesów: Get-Process oraz ps

```
Get-Process
```

```
ps aux
```

Obie komendy pokazują listę wszystkich uruchomionych programów razem z ich numerem procesu i zużyciem pamięci. `Get-Process` domyślnie sortuje po nazwie, `ps aux` po numerze PID, ale obie da się posortować inaczej.

Przykład:

```
PS C:\> Get-Process | Select-Object -First 2

Handles  NPM(K)  PM(K) Name
-------  ------  ----- ----
    412      21  45200 lsass
    198      12  18400 dns

[tomek@docker01 ~]$ ps aux | head -2
root  812  0.1  1.2 812400 45000 ?  Ssl  10:12  0:03 dockerd
tomek 2104 0.0  0.1  12800  3200 pts/0 R+  10:20  0:00 ps
```

PID = Process ID: numer procesu w systemie · **ps** = process status: pokaż procesy

TYLKO ODCZYT

Sprawdzenie stanu usługi: Get-Service oraz systemctl status

```
Get-Service -Name Spooler
```

```
systemctl status sshd
```

Obie komendy mówią, czy dana usługa jest uruchomiona, zatrzymana albo w ogóle nie istnieje na tym systemie. `systemctl status` dodatkowo pokazuje kilka ostatnich linii dziennika tej usługi, czego `Get-Service` nie robi.

Przykład:

```
PS C:\> Get-Service -Name Spooler

Status   Name     DisplayName
------   ----     -----------
Running  Spooler  Print Spooler

[tomek@docker01 ~]$ systemctl status sshd
● sshd.service - OpenSSH server daemon
     Active: active (running) since Sun 2026-09-27 08:02:11 CEST
```

systemctl = system control: sterowanie usługami systemd

TYLKO ODCZYT

Przeglądanie dziennika zdarzeń: Get-WinEvent oraz journalctl

```
Get-WinEvent -LogName System -MaxEvents 5
```

```
journalctl -n 5
```

Obie komendy pokazują najnowsze wpisy z dziennika systemowego, czyli zapis tego, co ostatnio działo się z systemem i usługami.

> **Pojęcie: systemd.** Systemd to program w Linuksie, który uruchamia usługi, pilnuje ich i prowadzi ich wspólny dziennik. `journalctl` czyta właśnie ten dziennik, tak jak Podgląd zdarzeń czyta dziennik Windows.

Przykład:

```
PS C:\> Get-WinEvent -LogName System -MaxEvents 2

TimeCreated       Id LevelDisplayName Message
-----------       -- ---------------- -------
27.09.2026 10:05   7036 Informacja   Usluga Spooler przeszla w stan dzialania

[tomek@docker01 ~]$ journalctl -n 2
wrz 27 10:05:02 docker01 sshd[2201]: Accepted publickey for tomek
```

journalctl = journal control: czytanie dziennika systemd

TYLKO ODCZYT

Sprawdzenie adresu IP i testowanie połączenia: Get-NetIPAddress, ping oraz ip, ping

```
Get-NetIPAddress -AddressFamily IPv4
ping dc01
```

```
ip addr show
ping -c 3 dc01
```

Pierwsza komenda w każdej parze pokazuje adresy IP przypisane do kart sieciowych tego komputera. Druga wysyła kilka pakietów testowych do wskazanej nazwy i mówi, czy tamten komputer odpowiada.

Przykład:

```
PS C:\> ping dc01


Odpowiedz z 10.20.10.11: bajtow=32 czas=1ms TTL=128


[tomek@docker01 ~]$ ping -c 3 dc01
64 bytes from 10.20.10.11: icmp_seq=1 ttl=128 time=0.912 ms
```

IP = Internet Protocol: adres komputera w sieci; w elektryce: stopień ochrony obudowy (Ingress Protection) · **ping** = Packet InterNet Groper: sprawdź, czy host odpowiada w sieci i jak długo trwa odpowiedź · **ip** = narzędzie do adresów, tras i interfejsów sieciowych

TYLKO ODCZYT

Sprawdzenie miejsca na dysku: Get-Volume oraz df

```
Get-Volume
```

```
df -h /docker
```

Obie komendy pokazują, ile miejsca zajęte jest na dysku, a ile zostało wolnego. `df -h` w Linuksie pokazuje wynik w megabajtach i gigabajtach zamiast surowych bajtów, dzięki fladze `-h`.

Przykład:

```
PS C:\> Get-Volume -DriveLetter C


DriveLetter FileSystem Size    SizeRemaining
----------- ---------- ----    -------------
C           NTFS       200 GB  74 GB


[tomek@docker01 ~]$ df -h /docker
/dev/mapper/vgDocker-lvDocker  100G  61G  40G  61% /docker
```

df = disk free: ile miejsca wolnego na dyskach

ZMIENIA SYSTEM

Instalacja programu: winget oraz dnf

```
winget install NazwaPakietu
```

```
sudo dnf install NazwaPakietu
```

> **Pojęcie: menedżer pakietów.** Menedżer pakietów to program, który pobiera, instaluje i aktualizuje gotowe programy z zaufanego repozytorium, zamiast szukać instalatora w internecie samodzielnie. Winget robi to w Windows, dnf w Rocky Linux i innych systemach z rodziny EL.

Przykład:

```
PS C:\> winget install Git.Git
Znaleziono Git.Git [Git]
Pobrano.

[tomek@docker01 ~]$ sudo dnf install git
Installed:
  git-2.43.0-1.el9.x86_64
```

EL = Enterprise Linux: rodzina RHEL i jej klonów (Rocky, Alma, Oracle Linux) · **winget** = Windows Package Manager: menedżer pakietów do instalacji programów w Windows · **dnf** = Dandified YUM: menedżer pakietów RHEL/Rocky

ZMIENIA SYSTEM

Ponowne uruchomienie usługi: Restart-Service oraz systemctl restart

```
Restart-Service -Name Spooler
```

```
sudo systemctl restart sshd
```

Obie komendy zatrzymują usługę i od razu uruchamiają ją ponownie, bez ponownego uruchamiania całego systemu. Po restarcie warto sprawdzić stan usługi tą samą komendą, którą sprawdzaliśmy go wcześniej.

Przykład:

```
PS C:\> Restart-Service -Name Spooler
PS C:\> Get-Service -Name Spooler

Status   Name
------   ----
Running  Spooler

[tomek@docker01 ~]$ sudo systemctl restart sshd
[tomek@docker01 ~]$ systemctl is-active sshd
active
```

systemctl = system control: sterowanie usługami systemd

Z życia wzięte

Z ŻYCIA WZIĘTE

Tomek dostaje zadanie sprawdzenia obciążenia na docker01 i odruchowo szuka ikony w prawym dolnym rogu ekranu, tak jak otwierał Menedżera zadań na swoim laptopie. Na serwerze bez monitora nie ma paska zadań ani ikon, jest tylko czarne okno terminala po SSH. Podpowiadamy mu polecenie `top`, które w Linuksie pokazuje to samo, co Menedżer zadań w Windows: listę procesów i ich zużycie zasobów na żywo. Po tym dniu Tomek zaczyna prowadzić własną notatkę z odpowiednikami komend.

Jak to działa

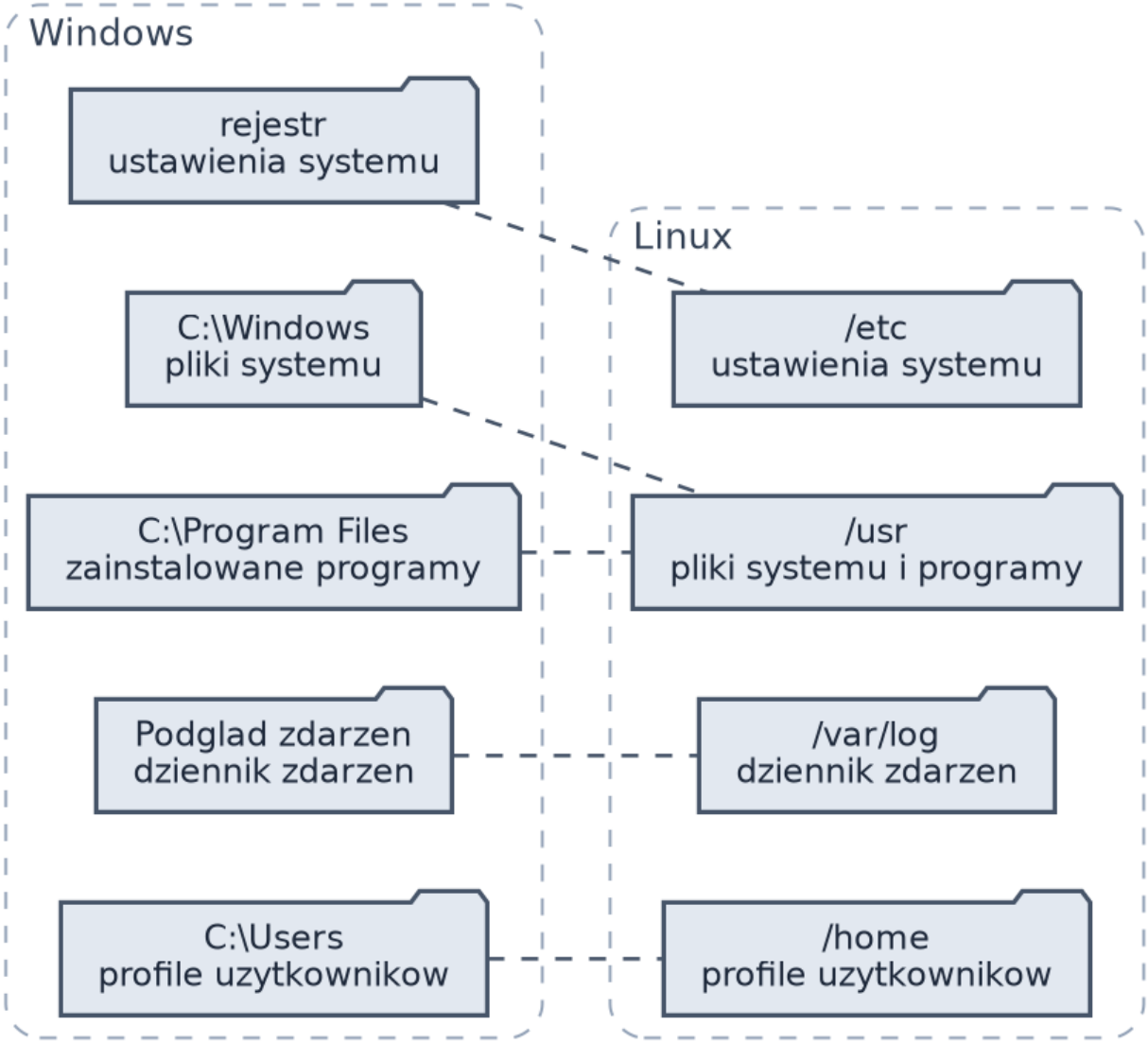
Rozwiń: jak to działa pod spodem

Poniższa tabela zbiera odpowiedniki najczęstszych czynności administratora w obu systemach. Nie trzeba się jej uczyć na pamięć, wracamy do niej, gdy zapomnimy jednej strony równania.

Co chcemy	Windows	Linux
Gdzie jesteśmy	`Get-Location`	`pwd`
Lista plików	`Get-ChildItem`	`ls -la`

Co chcemy	Windows	Linux
Podgląd pliku	`Get-Content`	`cat`, `less`
Szukanie tekstu w pliku	`Select-String`	`grep`
Kopiowanie pliku	`Copy-Item`	`cp`
Przenoszenie i zmiana nazwy	`Move-Item`, `Rename-Item`	`mv`
Usuwanie pliku	`Remove-Item`	`rm`
Tworzenie katalogu	`New-Item -ItemType Directory`	`mkdir`
Procesy	`Get-Process`	`ps aux`, `top`
Usługi	`Get-Service`	`systemctl status`
Restart usługi	`Restart-Service`	`systemctl restart`
Dziennik zdarzeń	`Get-WinEvent`	`journalctl`
Adres IP	`Get-NetIPAddress`	`ip addr`
Test połączenia	`ping`	`ping`
Miejsce na dysku	`Get-Volume`	`df -h`
Instalacja programu	`winget install`	`dnf install`
Edytor tekstu w terminalu	brak odpowiednika w CLI, zwykle Notatnik	`nano`, `vim`
Użytkownicy systemu	`Get-LocalUser`	`getent passwd`, `id`
Uprawnienia do pliku	zakładka Zabezpieczenia (NTFS, ACL)	`chmod`, `chown`
Konfiguracja systemu	rejestr	pliki w `/etc`

> **Pojęcie: rejestr systemu Windows.** Rejestr to jedna wielka baza ustawień całego systemu i zainstalowanych programów, ukryta pod programem `regedit`. Linux nie ma jednego takiego miejsca, każdy program trzyma swoje ustawienia we własnym pliku tekstowym w katalogu `/etc`.



Ustawienia systemu, pliki i dzienniki zdarzeń: te same rzeczy pod innymi nazwami w Windows i Linuksie (kliknięcie otwiera pełny rozmiar)

Poza nazwami komend te dwa systemy różnią się też sposobem myślenia, nie tylko słownictwem. Windows nie rozróżnia wielkości liter w nazwach plików, więc `Raport.txt` i `raport.txt` to ten sam plik. Linux rozróżnia je zawsze, więc te dwie nazwy to dwa osobne pliki, co bywa źródłem pomyłek dla kogoś, kto zaczynał na Windows.

Windows dzieli ścieżki ukośnikiem wstecznym `\`, jak w `C:\Program Files`, Linux zwykłym ukośnikiem `/`, jak w `/usr/bin`. PowerShell w praktyce toleruje też zwykły ukośnik, ale bash nigdy nie zrozumie odwrotnego.

> **Pojęcie: powłoka.** Powłoka to program, przez który wpisujemy komendy i dostajemy odpowiedzi, w Windows to PowerShell albo starszy CMD, w Linuksie najczęściej bash.

Linux trzyma się zasady „wszystko jest plikiem": dysk, drukarka, a nawet lista procesów w `/proc` dają się odczytać tymi samymi komendami co zwykły plik tekstowy. Windows takiej jednej zasady nie ma, dysk czyta się inaczej niż rejestr, a rejestr inaczej niż dziennik zdarzeń.

> **Pojęcie: potok.** Potok, zapisywany znakiem `|`, przekazuje wynik jednej komendy jako wejście do drugiej. W PowerShellu w potoku płyną gotowe obiekty z nazwanymi polami, w bashu płynie zwykły tekst, który trzeba dopiero rozciąć komendami takimi jak `grep` czy `awk`.

Częste błędy

Objaw	Przyczyna	Co robimy
`cd Dokumenty` w bashu nie działa, choć folder Documents istnieje	Linux rozróżnia wielkość liter, `Dokumenty` to nie to samo co `documents`	wpisujemy nazwę dokładnie tak, jak pokazuje `ls`
Ścieżka z Windows wklejona do basha nie działa	ukośniki wsteczne `\` nie znaczą w bashu tego samego co w Windows	zamieniamy `\` na `/`
`ls` w PowerShellu działa, ale niektóre flagi Linuksa nie	`ls` w PowerShellu to alias `Get-ChildItem`, nie prawdziwy `ls`	używamy natywnych parametrów PowerShella albo Windows Terminal z prawdziwym bashem
Zmiana w rejestrze nie widoczna od razu	niektóre klucze rejestru wymagają restartu usługi albo systemu	sprawdzamy w dokumentacji, czy dana zmiana wymaga restartu
Brak uprawnień mimo bycia administratorem	w Linuksie zwykłe konto nie ma praw roota, trzeba `sudo`	dopisujemy `sudo` przed komendą wymagającą uprawnień

□□ Ćwiczenia

1. Kolega z zespołu Oli mówi, że na Linuksie zgubił się plik `Raport.txt`, bo `ls` pokazuje tylko `raport.txt`. Co się właściwie stało?

Odpowiedź

Nic się nie zgubiło, oba pliki to na Linuksie dwie różne nazwy, bo system rozróżnia wielkość liter. W Windows `Raport.txt` i `raport.txt` byłyby tym samym plikiem, na Linuksie trzeba wpisać nazwę dokładnie tak, jak pokazuje `ls`.

1. Chcemy sprawdzić, czy usługa SSH działa na `docker01`, tak jak sprawdzalibyśmy usługę drukowania na Windows przez `Get-Service`. Jakiej komendy użyjemy i co dodatkowo zobaczymy w jej wyniku?

Odpowiedź

Użyjemy `systemctl status sshd`. Oprócz samego stanu usługi (uruchomiona albo zatrzymana) zobaczymy też kilka ostatnich linii jej dziennika, czego `Get-Service` nie pokazuje.

1. Dlaczego wynik `Get-Process` w PowerShellu łatwiej dalej przetworzyć niż wynik `ps aux` w bashu?

Odpowiedź

PowerShell przekazuje w potoku gotowe obiekty z nazwanymi polami, więc można od razu odwołać się na przykład do pola z zużyciem pamięci. Bash przekazuje zwykły tekst, więc żeby wyciągnąć jedną kolumnę, trzeba go dodatkowo pociąć komendami takimi jak `awk` czy `cut`.

Revision #29
Created 2026-09-28 05:52:35 UTC by Tomasz Konieczny
Updated 2026-10-26 14:18:04 UTC by Tomasz Konieczny