Montowanie: mount, /etc/fstab, zasoby sieciowe

Poziom: średni **Dotyczy: RHEL/Rocky 9–10, różnice Debian/Ubuntu i SLES zaznaczone**

Czas: ~22 min

`mount` podpina system plików pod katalog, `/etc/fstab` pamięta tę decyzję na stałe. Ta strona pokazuje obie strony: ręczne montowanie na chwilę i trwałą konfigurację, także dla zasobów sieciowych — NFS i CIFS/SMB.

Skróty użyte na tej stronie (9)

Skrót	Co znaczy
CIFS	Common Internet File System: starsza nazwa protokołu współdzielenia plików Windows, dziś częściej nazywanego SMB, pojawia się w wynikach poleceń FreeIPA dotyczących zaufania z Active Directory
IP	Internet Protocol: adres komputera w sieci; w elektryce: stopień ochrony obudowy (Ingress Protection)
NFS	Network File System: udostępnianie katalogów po sieci w świecie Linuksa
PID	Process ID: numer procesu w systemie
RHEL	Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa
SCSI	Small Computer System Interface: standard podłączania dysków
SLES	SUSE Linux Enterprise Server: płatna dystrybucja Linuksa firmy SUSE, konkurent Rocky/RHEL
UUID	Universally Unique Identifier: długi, praktycznie niepowtarzalny identyfikator
WMS	Warehouse Management System: system zarządzania magazynem

Fiszka

ZMIENIA SYSTEM

Wypisanie wszystkiego, co jest zamontowane: mount

```
mount
```

Bez argumentów wypisujemy każdy zamontowany system plików w systemie, razem z punktem montowania i opcjami. Lista bywa długa — obejmuje też wirtualne systemy plików jądra (`proc`, `sysfs`, `tmpfs`), nie tylko dyski.

Przykład:

```
/dev/mapper/vgOS-lvOS on / type xfs (rw,relatime,attr2,inode64)
/dev/sda1 on /boot type xfs (rw,relatime,attr2,inode64)
tmpfs on /run type tmpfs (rw,nosuid,nodev,size=1618132k)
bkp01.zielona.example:/export/backup on /backup type nfs4 (rw,relatime,vers=4.2,_netdev)
```

mount = podepnij system plików z urządzenia albo obrazu pod wskazany katalog

TYLKO ODCZYT

Wyświetlenie tego samego czytelniej: findmnt

```
findmnt
```

Prezentujemy te same informacje co `mount`, ale jako drzewo, z widoczną hierarchią punktów montowania jeden pod drugim. Wygodniejsze przy dużej liczbie zamontowanych zasobów.

Przykład:

```
TARGET                         SOURCE                                  FSTYPE OPTIONS
/                              /dev/mapper/vgOS-lvOS                   xfs    rw,relatime
├─/boot                        /dev/sda1                               xfs    rw,relatime
└─/backup                      bkp01.zielona.example:/export/backup nfs4   rw,_netdev
```

findmnt = pokaż zamontowane systemy plików w czytelnej formie drzewa

ZMIENIA SYSTEM

Montowanie ręczne, tymczasowe: mount

```
mount /dev/sdX /mnt
```

(gdzie `/dev/sdX` to urządzenie blokowe, które chcemy podpiąć)

Podpinamy system plików pod wskazany katalog na czas do najbliższego odmontowania albo restartu. Nie trafia do `/etc/fstab` — po restarcie zniknie. Dobre do sprawdzenia dysku, złe jako rozwiązanie docelowe.

mount = podepnij system plików z urządzenia albo obrazu pod wskazany katalog

ZMIENIA SYSTEM

Montowanie z opcjami: -o ro,noexec

```
mount -o ro,noexec /dev/sdX /mnt
```

`-o` przyjmuje listę opcji montowania rozdzieloną przecinkami. `ro` montuje tylko do odczytu (przydatne przy odzyskiwaniu danych z uszkodzonego dysku — zero ryzyka dopisania czegoś przez system plików w trakcie fsck), `noexec` blokuje uruchamianie plików wykonywalnych z tego punktu montowania — częste zabezpieczenie na `/tmp` i katalogach z uploadami.

mount = podepnij system plików z urządzenia albo obrazu pod wskazany katalog

TYLKO ODCZYT

Odmontowanie: umount

```
umount /mnt
```

Odwrotność `mount`. Jeśli w katalogu (albo pod nim) coś nadal ma otwarty plik, komenda odmawia — patrz *Częste błędy*.

umount = unmount: odmontowuje system plików; nie działa, dopóki jakiś proces trzyma otwarte pliki na tym woluminie

TYLKO ODCZYT

Odmontowanie leniwe: -l

```
umount -l /mnt
```

„Lazy unmount” odłącza punkt montowania z drzewa katalogów natychmiast, ale faktyczne zwolnienie zasobu odkłada do momentu, aż przestanie być używany przez ostatni proces.

Rozwiązuje objaw, nie przyczynę — dobre w awaryjnej sytuacji, złe jako nawyk, bo maskuje proces, który i tak trzeba będzie znaleźć i zamknąć.

umount = unmount: odmontowuje system plików; nie działa, dopóki jakiś proces trzyma otwarte pliki na tym woluminie

TYLKO ODCZYT

Sprawdzenie, kto trzyma punkt montowania: fuser -vm

```
fuser -vm /mnt
```

Wypisujemy procesy, które mają otwarty plik, katalog roboczy albo uruchomiony program z podanego punktu montowania. Pierwszy odruch przy `target is busy`.

Przykład:

```
                     USER        PID ACCESS COMMAND
/mnt:                root       1842 ..c.. bash
                      root       1901 f.... tail
```

Litery w kolumnie `ACCESS` mówią, jak proces trzyma zasób: `c` — katalog roboczy, `f` — otwarty plik.

TYLKO ODCZYT

Wyświetlenie pełnej listy otwartych plików: lsof +D

```
lsof +D /mnt
```

`fuser` mówi, kto trzyma punkt montowania; `lsof +D` mówi dokładnie, **który plik** ten proces trzyma otwarty, rekurencyjnie przeszukując cały katalog. Wolniejsze na dużych drzewach katalogów, ale precyzyjniejsze.

Przykład:

```
COMMAND  PID USER   FD   TYPE DEVICE SIZE/OFF    NODE NAME
tail    1901 root    3r   REG  253,1  1048576  131074 /mnt/log/app.log
```

TYLKO ODCZYT

Wpis w `/etc/fstab` — sześć pól

Przykład:

```
UUID=3a1f2c9e-... /data ext4 defaults,nofail 0 2
```

Pole	Znaczenie
1. urządzenie	co montować: `/dev/sdX`, `UUID=...`, `LABEL=...`, adres sieciowy (`serwer:/eksport`)
2. punkt montowania	katalog, pod który zasób ma się podpiąć
3. typ systemu plików	`xfs`, `ext4`, `nfs`, `nfs4`, `cifs`, `swap`...
4. opcje	lista rozdzielona przecinkami: `defaults`, `ro`, `noexec`, `_netdev`, `nofail`...
5. dump	flaga dla starego narzędzia `dump` (0 — wyłącz, w praktyce prawie zawsze 0)
6. fsck pass	kolejność sprawdzania `fsck` przy starcie: 0 — nie sprawdzaj, 1 — system root, 2 — pozostałe

UUID = Universally Unique Identifier: długi, praktycznie niepowtarzalny identyfikator

ZMIENIA SYSTEM

Zamontowanie wszystkiego z fstab bez restartu: mount -a

```
mount -a
```

Przechodzimy przez cały `/etc/fstab` i montujemy każdy wpis, który jeszcze nie jest zamontowany. Standardowy sposób przetestowania nowego wpisu — jeśli `mount -a` przechodzi bez błędu, wpis jest poprawny składniowo i praktycznie.

mount = podepnij system plików z urządzenia albo obrazu pod wskazany katalog

ZMIENIA SYSTEM

Przeładowanie systemd po zmianie fstab: systemctl daemon-reload

```
systemctl daemon-reload
```

systemd generuje jednostki `.mount` na podstawie `/etc/fstab` przy każdym starcie, ale trzyma też ich zapamiętaną wersję w pamięci działającego demona. Po ręcznej edycji `fstab` i przed próbą zarządzania nowym wpisem bezpośrednio przez `systemctl` (np. `systemctl start`

`data.mount`) trzeba kazać systemd przeczytać plik na nowo.

systemctl = system control: sterowanie usługami systemd

ZMIENIA SYSTEM

Ręczne montowanie zasobu NFS: mount -t nfs

```
mount -t nfs bkp01.zielona.example:/export/backup /backup
```

`-t nfs` (albo `nfs4` dla wersji 4 protokołu) mówi jądru, jakiego klienta systemu plików użyć. Adres przed dwukropkiem to serwer, ścieżka po dwukropku to eksportowany katalog po stronie serwera — nie ścieżka lokalna.

NFS = Network File System: udostępnianie katalogów po sieci w świecie Linuksa · **mount** = podepnij system plików z urządzenia albo obrazu pod wskazany katalog

TYLKO ODCZYT

Wpis NFS w `/etc/fstab` z `_netdev`

Przykład:

```
bkp01.zielona.example:/export/backup /backup nfs4 defaults,_netdev,nofail 0 0
```

`_netdev` mówi systemd, że to zasób sieciowy — jednostka montująca poczeka, aż sieć będzie gotowa, zamiast próbować (i failować) w pierwszych sekundach rozruchu, jeszcze przed uzyskaniem adresu IP.

NFS = Network File System: udostępnianie katalogów po sieci w świecie Linuksa · **IP** = Internet Protocol: adres komputera w sieci; w elektryce: stopień ochrony obudowy (Ingress Protection)

ZMIENIA SYSTEM

Montowanie zasobu Windows/Samba: mount -t cifs -o credentials=

```
mount -t cifs -o credentials=/etc/samba/creds-backup //bkp01.zielona.example/backup
/mnt/backup
```

CIFS/SMB to protokół udostępniania plików rodem z Windows, obsługiwany też przez Sambę na Linuksie. Zamiast podawać login i hasło wprost w linii poleceń (widoczne w historii powłoki i w `ps aux`), wskazuje się plik z poświadczeniami przez `credentials=`.

□□ **CIFS** = Common Internet File System: starsza nazwa protokołu współdzielenia plików Windows, dziś częściej nazywanego SMB, pojawia się w wynikach poleceń FreeIPA dotyczących zaufania z Active Directory · **mount** = podepnij system plików z urządzenia albo obrazu pod wskazany katalog

□□ ZMIENIA SYSTEM

Utworzenie pliku poświadczeń CIFS: chmod 600

```
cat > /etc/samba/creds-backup <<'EOF'
username=backupuser
password=zmien-to-haslo
domain=EXAMPLE
EOF
chmod 600 /etc/samba/creds-backup
```

Plik z hasłem musi być czytelny wyłącznie dla właściciela. Bez `chmod 600` każdy lokalny użytkownik z dostępem do odczytu tego pliku poznaje hasło do zasobu sieciowego — `mount` nie sprawdza uprawnień pliku za nas.

□□ **CIFS** = Common Internet File System: starsza nazwa protokołu współdzielenia plików Windows, dziś częściej nazywanego SMB, pojawia się w wynikach poleceń FreeIPA dotyczących zaufania z Active Directory · **cat** = concatenate: sklej i wypisz pliki · **chmod** = change mode: zmień prawa dostępu

□□ TYLKO ODCZYT

Wypisanie identyfikatorów dysków: blkid

```
blkid
```

Wypisujemy UUID, etykietę (LABEL) i typ systemu plików dla każdego wykrytego urządzenia blokowego. Źródło wartości, które wstawiamy do `/etc/fstab` zamiast nazwy `/dev/sdX`.

Przykład:

```
/dev/sda1: UUID="3a1f2c9e-8b4d-4e21-9c77-1f6a2d8e9b10" TYPE="xfs" PARTUUID="..."
/dev/sdb1: UUID="7c2e91aa-44f0-4b8d-9a11-002233445566" TYPE="ext4"
```

□□ **UUID** = Universally Unique Identifier: długi, praktycznie niepowtarzalny identyfikator · **blkid** = pokaż UUID i typ systemu plików na urządzeniu blokowym

ZMIENIA SYSTEM

Montowanie ad hoc przez systemd: systemd-mount

```
systemd-mount --no-block /dev/sdX /mnt
```

Alternatywa dla `mount` do jednorazowego, tymczasowego podpięcia — tworzy jednostkę `.mount` w locie, zarządzaną przez systemd, widoczną w `systemctl status`. Przydatne, gdy montowanie ma być widoczne i sterowalne jak każda inna jednostka systemd, ale nie chcemy na stałe wpisywać go do `fstab`.

> **Uważaj**
>
> - Kolejność wpisu w `/etc/fstab` w praktyce nie ma znaczenia dla `mount -a`, ale ma znaczenie logiczne — katalogi nadrzędne warto trzymać przed podkatalogami, dla czytelności.
> - `nofail` w opcjach nie jest domyślne — bez niego błąd montowania wstrzymuje cały rozruch systemu.
> - Plik poświadczeń CIFS bez `chmod 600` to nie błąd składniowy, tylko cichy wyciek hasła.

CIFS = Common Internet File System: starsza nazwa protokołu współdzielenia plików Windows, dziś częściej nazywanego SMB, pojawia się w wynikach poleceń FreeIPA dotyczących zaufania z Active Directory

Z życia wzięte

Z ŻYCIA WZIĘTE

Poniedziałek, 8:00. Serwer app01 nie wstaje po planowanym restarcie, bo w fstab ktoś dopisał udział sieciowy z fs01 bez opcji, która pozwala systemowi ruszyć dalej, gdy sieć jeszcze nie działa. Marek z magazynu dzwoni, że WMS też nie odpowiada, więc Tomek poprawia wpis i dopisuje _netdev, zanim ktokolwiek zdąży zgłosić to jako awarię.

Po co to istnieje

> **Pojęcie: montowanie i punkt montowania.** W Linuksie nie ma liter dysków jak w Windows (`C:`, `D:`) — każdy system plików musi zostać *dołączony* do jednego, wspólnego drzewa katalogów zaczynającego się od `/`. Ten akt dołączenia nazywa się montowaniem, a katalog, pod który dany system plików się podpina, punktem montowania. Zanim zamontujesz dysk pod `/data`, katalog `/data` jest zwykłym, pustym (albo niepustym — patrz uwaga niżej) katalogiem na systemie plików root. Po zamontowaniu wszystko, co zapiszesz pod `/data`, faktycznie ląduje na tamtym dysku, a nie na dysku systemowym.

Punkt montowania nie musi być pusty przed zamontowaniem — jeśli zawiera pliki, stają się niewidoczne (nie usunięte, tylko zasłonięte) na czas, gdy coś innego jest tam zamontowane. Stąd podstępna pomyłka: administrator kopiuje pliki do katalogu bez jeszcze niczego zamontowanego, a po restarcie te pliki „znikają” — leżą bezpiecznie na dysku systemowym, tylko przestały być widoczne pod tą ścieżką.

Analogia

Punkt montowania to gniazdko elektryczne w ścianie. Samo gniazdko nic nie robi — dopiero podłączenie urządzenia (pralki, lodówki) sprawia, że zaczyna przez nie płynąć prąd o właściwościach tego urządzenia. Katalog `/data` to takie gniazdko: puste, dopóki nic w nie nie „wpięto”. `mount /dev/sdb1 /data` to fizyczne podłączenie wtyczki — od tej pory wszystko, co dzieje się „w gniazdku”, obsługuje podłączone urządzenie.

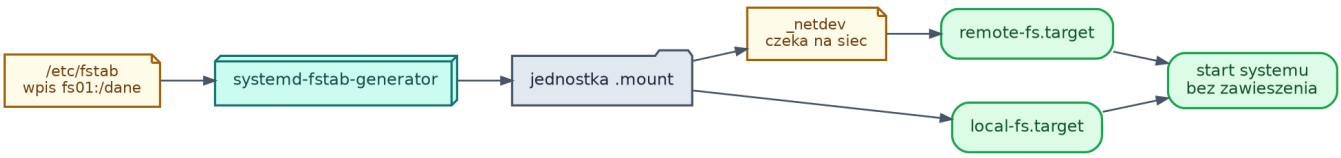
`/etc/fstab` to instrukcja dla elektryka na wypadek przerwy w dostawie prądu: „gdy prąd wróci, podłącz z powrotem dokładnie te urządzenia do dokładnie tych gniazdek”. Pomyłka w numerze gniazdka albo podłączenie urządzenia, które akurat nie jest gotowe (zasób sieciowy, zanim sieć ruszyła), może zostawić całe mieszkanie bez prądu — to dokładnie mechanizm błędnego wpisu w `fstab`, opisany niżej.

Jak to działa

Rozwiń: jak to działa pod spodem

> **Pojęcie: system plików sieciowy.** Zwykły system plików (`ext4`, `xfs`) czyta i zapisuje dane bezpośrednio z lokalnego urządzenia blokowego. System plików sieciowy robi to samo z punktu widzenia aplikacji — programy widzą zwykłe pliki i katalogi — ale pod spodem każda operacja (odczyt, zapis, listowanie katalogu) jest w rzeczywistości zapytaniem sieciowym do zdalnego serwera. NFS (Network File System, natywny dla

Uniksa/Linuksa) i CIFS/SMB (protokół z rodziny Windows, obsługiwany po stronie serwera przez Sambę) to dwa najpopularniejsze przykłady. Konsekwencja tej różnicy: system plików sieciowy może w każdej chwili przestać odpowiadać z powodów, które nie mają nic wspólnego z samym serwerem plików — awaria sieci, przeciążenie, restart serwera — i klient musi jakoś sobie z tym poradzić (patrz *Częste błędy*, „stale file handle”).



Wpis w fstab trafia przez generator systemd do jednostki mount, która czeka na odpowiedni target. (kliknięcie otwiera pełny rozmiar)

Pojęcie: opcje montowania. Opcje w czwartym polu `fstab` (i w `-o` przy `mount`) kontrolują, jak system plików ma się zachowywać po zamontowaniu — niezależnie od tego, co jest na nim fizycznie zapisane. `ro` wymusza tylko odczyt na poziomie jądra, nawet jeśli użytkownik ma uprawnienia do zapisu w metadanych plików. `noexec` blokuje wykonywanie plików, `nosuid` ignoruje bit `setuid`, `_netdev` i `nofail` (opisane niżej) kontrolują zachowanie przy rozruchu. Poniższe dwie opcje są tu kluczowe, bo to one decydują, czy błędny wpis w `fstab` zablokuje cały system, czy tylko jeden katalog:

- `_netdev` — mówi systemd, że urządzenie źródłowe jest osiągalne dopiero po podniesieniu sieci. Jednostka `.mount` z tą opcją czeka na `network-online.target`, zamiast próbować zamontować się w pierwszych sekundach rozruchu, gdy interfejs sieciowy jeszcze nie ma adresu.
- `nofail` — mówi systemd, żeby nieudane zamontowanie tego konkretnego wpisu **nie** blokowało reszty procesu startowego. Bez `nofail` błąd montowania jednego wpisu w `fstab` potrafi zatrzymać cały rozruch systemu w trybie ratunkowym, nawet jeśli problem dotyczy pojedynczego, mało istotnego katalogu.

Pojęcie: jednostka mount w systemd. Współczesny systemd nie czyta `/etc/fstab` bezpośrednio w czasie działania — generator `systemd-fstab-generator` przetwarza go raz, przy każdym starcie systemu (i po `daemon-reload`), i tworzy z niego jednostki typu `.mount`, jedną na wpis, nazwane od ścieżki punktu montowania (np. `/data` staje się jednostką `data.mount`). Dzięki temu montowanie z `fstab` korzysta z tego samego mechanizmu zależności co reszta systemd — może czekać na inne jednostki, mieć limit czasu, być widoczne w `systemctl status`. To także wyjaśnia, dlaczego po ręcznej edycji `fstab`

> czasem trzeba `daemon-reload`: systemd operuje na wygenerowanych jednostkach, nie na samym pliku tekstowym.

Krok po kroku

1. Sprawdź, co jest zamontowane teraz.

```
findmnt
```

2. Znajdź identyfikator nowego dysku — po UUID, nie po nazwie urządzenia.

```
blkid /dev/sdb1
```

```
/dev/sdb1: UUID="7c2e91aa-44f0-4b8d-9a11-002233445566" TYPE="ext4"
```

3. Zamontuj ręcznie, żeby przetestować, zanim wpiszesz cokolwiek na stałe.

```
mkdir -p /data
mount /dev/sdb1 /data
df -h /data
```

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdb1       100G   61M   95G   1% /data
```

4. Odmontuj i dodaj trwały wpis do `/etc/fstab`, po UUID.

```
umount /data
echo 'UUID=7c2e91aa-44f0-4b8d-9a11-002233445566 /data ext4 defaults,nofail 0 2' >> /etc/fstab
```

5. Przetestuj wpis komendą `mount -a`, bez restartu.

```
mount -a
df -h /data
```

6. Przykład NFS — montowanie zasobu sieciowego z `_netdev`.

```
mkdir -p /backup
mount -t nfs bkp01.zielona.example:/export/backup /backup
```

```
echo 'bkp01.zielona.example:/export/backup /backup nfs4 defaults,_netdev,nofail 0 0' >>
/etc/fstab
mount -a
df -h /backup
```

7. Przykład CIFS/SMB — montowanie z plikiem poświadczeń.

```
cat > /etc/samba/creds-backup <<'EOF'
username=backupuser
password=zmien-to-haslo
domain=EXAMPLE
EOF
chmod 600 /etc/samba/creds-backup

mkdir -p /mnt/backup-smb
echo '//bkp01.zielona.example/backup /mnt/backup-smb cifs credentials=/etc/samba/creds-
backup,_netdev,nofail 0 0' >> /etc/fstab
mount -a
```

8. Po każdej edycji `fstab`, zanim spróbujesz zarządzać wpisem przez `systemctl`.

```
systemctl daemon-reload
```

Częste błędy

Montowanie po nazwie `/dev/sdX` zamiast po UUID.

```
mount: /data: special device /dev/sdb1 does not exist.
```

Kolejność, w jakiej jądro wykrywa dyski przy starcie, nie jest gwarantowana — zależy od kolejności inicjalizacji kontrolerów, dodania nowego dysku, a czasem zwykłego przypadku. Dysk, który dziś jest `/dev/sdb`, po dodaniu kolejnego kontrolera SCSI może następnym razem wystartować jako `/dev/sdc`. UUID jest przypisany na stałe do systemu plików w momencie jego utworzenia (`mkfs`) i nie zmienia się niezależnie od tego, pod jaką nazwą urządzenia dysk akurat się pojawi. Naprawa: zawsze `UUID=...` albo `LABEL=...` w `fstab`, nigdy gołe `/dev/sdX` dla dysków innych niż tymczasowe testy ręczne.

Błędny wpis w `fstab` bez `nofail` blokuje cały rozruch.

```
[ TIME ] Timed out waiting for device /dev/sdb1.
[DEPEND] Dependency failed for /data.
[DEPEND] Dependency failed for Local File Systems.
Give root password for maintenance
(or press Control-D to continue):
```

Literówka w UUID, dysk fizycznie odłączony, albo zasób sieciowy chwilowo niedostępny — bez `nofail` systemd traktuje nieudane zamontowanie jako awarię blokującą dalszy rozruch i ląduje w trybie konserwacji z prośbą o hasło roota. Naprawa doraźna: zalogować się w trybie konserwacji, poprawić albo usunąć wadliwy wpis, `systemctl default` albo restart. Naprawa systemowa: dodawać `nofail` do wpisów, które nie są absolutnie krytyczne dla działania systemu.

Brak `_netdev` na zasobie sieciowym.

```
[ TIME ] Timed out waiting for device bkp01.zielona.example:/export/backup.
```

Bez `_netdev` systemd próbuje zamontować zasób sieciowy razem z lokalnymi dyskami, na bardzo wczesnym etapie rozruchu — zanim interfejs sieciowy w ogóle dostanie adres IP. Efekt to kilkuminutowe oczekiwanie na timeout przy każdym starcie, a na zasobach bez `nofail` — pełne zablokowanie rozruchu. Naprawa: dopisać `_netdev` do opcji montowania i `systemctl daemon-reload`.

`umount: target is busy`.

```
umount: /mnt: target is busy.
```

Jakiś proces ma otwarty plik, katalog roboczy albo działający program wewnątrz punktu montowania. Namierzenie winowajcy:

```
fuser -vm /mnt
lsof +D /mnt
```

```
                     USER        PID ACCESS COMMAND
/mnt:                root       1842 ..c.. bash
```

Terminal z powłoką, której katalogiem roboczym jest `/mnt` (albo podkatalog), wystarczy, żeby zablokować `umount`. Naprawa: zamknąć albo przenieść (`cd /`) tę powłokę, albo w ostateczności `kill` na wskazany PID, i dopiero potem `umount`. `umount -l` omija problem natychmiast, ale nie rozwiązuje go — proces nadal trzyma otwarty plik, tylko punkt montowania znika z widoku wcześniej.

„Stale file handle” po restarcie serwera NFS.

```
ls: cannot access '/backup': Stale file handle
```

Klient NFS trzyma uchwyt (file handle) do plików na serwerze. Jeśli serwer zostanie zrestartowany, albo eksport zmieni identyfikator wewnętrznie (np. po przebudowie systemu plików po stronie serwera), stary uchwyt przestaje być ważny, a klient o tym nie wie, dopóki nie spróbuje go użyć. Naprawa: odmontować i zamontować ponownie (`umount -f /backup && mount -a`) — czasem konieczne jest `umount -l`, jeśli zwykłe `umount` też zwraca błąd z powodu tego samego nieważnego uchwytu.

□□ Ćwiczenia

> Poniższe ćwiczenia korzystają z pliku obrazu dysku montowanego przez `losetup`, więc nie dotykają prawdziwych partycji. Same operacje `mount`/`umount` wymagają uprawnień administratora (`sudo`).

Ćwiczenie 1 — montowanie po UUID zamiast po nazwie urządzenia

```
cd /tmp
dd if=/dev/zero of=test.img bs=1M count=64
sudo losetup -fP test.img
LOOP=$(losetup -j test.img | cut -d: -f1)
sudo mkfs.ext4 -L testowy "$LOOP"
sudo blkid "$LOOP"
```

Zanotuj UUID z wyniku `blkid`, a potem zamontuj po UUID, nie po nazwie urządzenia:

```
sudo mkdir -p /mnt/test
UUID=$(sudo blkid -s UUID -o value "$LOOP")
sudo mount UUID="$UUID" /mnt/test
findmnt /mnt/test
```

Dlaczego w środowisku produkcyjnym ta różnica ma znaczenie, skoro tutaj `$LOOP` i UUID prowadzą do tego samego urządzenia?

Odpowiedź

Tutaj różnicy nie widać, bo `$LOOP` jest ustalane w tej samej sesji i nie zmienia się między poleceniami. W prawdziwym systemie z wieloma dyskami nazwa `/dev/sdX` zależy od kolejności wykrycia dysków przez jądro przy starcie i może się zmienić po dodaniu, usunięciu albo nawet wymianie kolejności portów kontrolera SCSI/SAS. UUID jest zapisany wewnątrz systemu plików w momencie jego utworzenia i nie zależy od tego, pod jaką nazwą urządzenia dysk się akurat pojawi.

Sprzątanie:

```
sudo umount /mnt/test
sudo losetup -d "$LOOP"
rm test.img
```

Ćwiczenie 2 — `target is busy` i namierzenie winowajcy

```
cd /tmp
dd if=/dev/zero of=test2.img bs=1M count=32
sudo losetup -fP test2.img
LOOP=$(losetup -j test2.img | cut -d: -f1)
sudo mkfs.ext4 "$LOOP"
sudo mkdir -p /mnt/test2
sudo mount "$LOOP" /mnt/test2
```

Otwórz drugi terminal (albo w tle) i zostań w zamontowanym katalogu:

```
( cd /mnt/test2 && sleep 60 ) &
```

Spróbuj odmontować od razu:

```
sudo umount /mnt/test2
```

Powinieneś dostać `target is busy`. Znajdź winowajcę i odmontuj poprawnie:

Odpowiedź

```
fuser -vm /mnt/test2
```

```
                     USER        PID ACCESS COMMAND
/mnt/test2:          root       ####  ..c.. sleep
```

Proces `sleep` uruchomiony w podpowłoce trzyma `/mnt/test2` jako katalog roboczy (`c` w kolumnie `ACCESS`). Po zakończeniu procesu (poczekaniu na koniec `sleep 60` albo `kill` na wskazany PID) `umount` przechodzi normalnie:

```
sudo umount /mnt/test2
sudo losetup -d "$LOOP"
rm test2.img
```

Pytania kontrolne

1. Dlaczego montowanie po UUID jest bezpieczniejsze niż po `/dev/sdX`, skoro oba sposoby na pierwszy rzut oka „działają”?

Odpowiedź

Nazwa `/dev/sdX` jest przypisywana dynamicznie przez jądro w kolejności wykrywania dysków przy starcie i może się zmienić po dodaniu, usunięciu albo przełożeniu dysku na inny port kontrolera. UUID jest zapisany wewnątrz metadanych systemu plików w momencie jego utworzenia i pozostaje ten sam niezależnie od tego, jaką nazwę urządzenia system nada dyskowi tym razem. Wpis w `fstab` oparty na `/dev/sdX` może więc pewnego dnia zacząć wskazywać na zupełnie inny, przypadkowy dysk.

2. Co dokładnie robi opcja `_netdev` i dlaczego jej brak na zasobie NFS/CIFS objawia się dopiero przy restarcie, a nie przy ręcznym `mount -a`?

Odpowiedź

`_netdev` mówi systemd, że dane urządzenie jest osiągalne dopiero po podniesieniu sieci, więc odpowiednia jednostka `.mount` czeka na `network-online.target`, zamiast próbować się zamontować natychmiast. Ręczne `mount -a` uruchamiane już po pełnym starcie systemu zawsze działa w warunkach, gdzie sieć jest gotowa, więc brak `_netdev` nie ujawnia się w tym scenariuszu. Problem pojawia się wyłącznie podczas rozruchu systemu, kiedy bez `_netdev` systemd próbuje zamontować zasób sieciowy razem z lokalnymi dyskami, zanim interfejs w ogóle dostanie adres IP.

3. Masz błędny wpis w `/etc/fstab` (literówka w UUID) na wpisie bez `nofail`. Jaki będzie efekt przy najbliższym restarcie i jak temu zapobiec na przyszłość?

Odpowiedź

System zatrzyma się na etapie montowania systemów plików, wyświetli `Timed out waiting for device` i wyląduje w trybie konserwacji z prośbą o hasło roota — nawet jeśli błędny wpis dotyczył drobnego, nieistotnego katalogu. Zapobieganie: dodawanie `nofail` do wpisów, które nie są krytyczne dla uruchomienia systemu, żeby błąd w jednym z nich skutkował tylko brakującym punktem montowania, a nie zablokowanym rozruchem całej maszyny.

4. `umount /mnt` zwraca `target is busy`, a Ty nie masz otwartego żadnego terminala w tym katalogu. Jak dalej szukać przyczyny?

Odpowiedź

`fuser -vm /mnt` pokaże wszystkie procesy trzymające punkt montowania, nie tylko powłoki z ustawionym katalogiem roboczym — także procesy z otwartym plikiem gdzieś w drzewie katalogów. Jeśli lista jest niejasna, `lsof +D /mnt` pokaże dokładnie, który plik dany proces trzyma otwarty. Dopiero gdy żadne z tych narzędzi nic nie pokazuje, a `umount` nadal odmawia, warto sprawdzić, czy katalog nie jest przypadkiem bind-mountowany gdzie indziej albo czy nie trzyma go proces w innym przestrzeni nazw (kontener).

Powiązane: Zmiana nazwy grupy wolumenów i wolumenu logicznego · pvmove — przenoszenie danych między dyskami · Uprawnienia plików: chmod, chown, umask

Źródła: `man 8 mount` · `man 5 fstab` · `man 8 findmnt` · `man 1 fuser` · `man 8 systemd-mount` · `man 5 systemd.mount`

Revision #16
Created 2026-09-17 21:43:10 UTC by Tomasz Konieczny
Updated 2026-10-03 19:08:51 UTC by Tomasz Konieczny