Tydzień 1 do EX200: narzędzia, skrypty, start systemu i magazyn lokalny

Poziom: podstawy **Dotyczy: RHEL 10 (rhlab01 i rhlab02), cele EX200**

Czas: 8 dni po 1,5 do 2 godzin

Dni 1 do 8 programu: powłoka i dokumentacja, pliki i archiwa, skrypty, start systemu i hasło roota, procesy i dzienniki, partycje GPT i LVM, systemy plików, NFS i autofs. Każdy dzień ma scenkę, cele egzaminu, strony z bazy do przeczytania i ćwiczenia z odpowiedziami.

ZAŁOŻENIA SCENARIUSZA

ćwiczenia robimy w laboratorium ze strony Plan 30 dni i laboratorium RHEL w Zielonej: `rhlab01` i `rhlab02` na RHEL 10 w sieci 10.20.150.0/24, konto `tomek` z `sudo`, dwa dodatkowe dyski po 5 GB na każdej maszynie. Nazwy dysków (`sdb`, `sdc`) zależą od kontrolera, więc przed każdym ćwiczeniem z dyskami sprawdzamy `lsblk`.

Skróty użyte na tej stronie (16)

Skrót	Co znaczy
EX200	numer egzaminu praktycznego RHCSA, na którym kandydat konfiguruje żywy system RHEL bez dostępu do internetu
GB	Gigabyte: miliard bajtów, jednostka pojemności pamięci i plików, tu najczęściej rozmiar modelu albo pamięci karty graficznej
GiB	gibibajt: 1024 MiB, czyli około 1,07 GB; tej jednostki używa dokumentacja MailCow przy wymaganiach pamięci
GPT	GUID Partition Table: nowoczesny sposób dzielenia dysku na partycje
GRUB	Grand Unified Bootloader: program, który jako pierwszy startuje po włączeniu komputera i podaje jądru, co ma uruchomić
GUID	Globally Unique Identifier: unikalny w skali globalnej identyfikator obiektu, tu konkretnego protektora klucza BitLockera

Skrót	Co znaczy
LVM	Logical Volume Manager: warstwa w Linuksie składająca dyski w elastyczne wolumeny
MBR	Master Boot Record: stary sposób dzielenia dysku na partycje, do 2 TB
NFS	Network File System: udostępnianie katalogów po sieci w świecie Linuksa
RH134	kurs Red Hat o administracji systemem RHEL na poziomie RHCSA, kontynuacja RH124
RHEL	Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa
SSH	Secure Shell: szyfrowane zdalne logowanie do serwera
UID	User ID: numer użytkownika w Linuksie
UUID	Universally Unique Identifier: długi, praktycznie niepowtarzalny identyfikator
VFAT	Virtual FAT: sterownik systemu plików z rodziny FAT, używany na pendrive'ach i partycjach rozruchowych EFI
XFS	system plików stworzony przez Silicon Graphics, domyślny w Rocky Linux i RHEL

Fiszka

Komendy wykonujemy w bash jako `tomek` z `sudo`, na `rhlab01`, chyba że ćwiczenie mówi inaczej. Nazwy takie jak `NazwaPliku` czy `NazwaDysku` zastępujemy własnymi. Cele egzaminu pochodzą ze strony Red Hat o egzaminie EX200 (RHEL 10), stan na 10 października 2026.

□□ TYLKO ODCZYT

Szukanie w dokumentacji po słowie kluczowym: man -k

```
man -k NazwaTematu
```

Przeszukuje opisy stron `man`. Gdy wynik jest pusty, indeks trzeba odbudować komendą `sudo mandb`. Strona z działu 5 opisuje format pliku, na przykład `man 5 fstab`.

□□ **man** = manual: pokaż podręcznik systemowy do polecenia

□□ ZMIENIA SYSTEM

Zapis wyniku i błędów do osobnych plików: >, 2>

```
find /etc -name "*.conf" > wynik.txt 2> bledy.txt
```

Strumień wyniku idzie do `wynik.txt`, a komunikaty o błędach, na przykład o braku uprawnień, do `bledy.txt`. Opisuje to strona Przekierowania: >, >>, 2>, &>, <, tee.

find = przeszukaj drzewo katalogów według nazwy, typu, daty albo rozmiaru

TYLKO ODCZYT

Wyszukanie aktywnych linii w plikach konfiguracyjnych: grep -rEv

```
grep -rEv '^[[:space:]]*(#|$)' /etc/ssh/sshd_config /etc/ssh/sshd_config.d/
```

Pomija komentarze i puste linie, więc zostaje to, co naprawdę działa. Odwrócenie wzorca opcją `-v` jest jednym z najczęściej używanych trików przy czytaniu plików konfiguracyjnych.

grep = global regular expression print: szukaj wzorca w tekście

TYLKO ODCZYT

Utworzenie archiwum skompresowanego: tar -czf

```
sudo tar -czf /tmp/NazwaArchiwum.tar.gz -C /etc ssh
```

Opcja `-z` kompresuje gzipem, `-j` bzip2, `-J` xz. Opcja `-C` zmienia katalog, z którego liczone są ścieżki w archiwum.

tar = tape archive: archiwum plików (kiedyś na taśmę)

TYLKO ODCZYT

Rozpakowanie jednego pliku z archiwum: tar -xzf

```
tar -xzf /tmp/NazwaArchiwum.tar.gz -C /tmp/NazwaKatalogu ssh/sshd_config
```

Listę zawartości bez rozpakowywania pokazuje `tar -tzf`. Katalog docelowy z opcji `-C` musi już istnieć.

tar = tape archive: archiwum plików (kiedyś na taśmę)

ZMIENIA SYSTEM

Utworzenie dowiązania symbolicznego: ln -s

```
ln -s /etc/hostname ~/NazwaDowiazania
```

Dowiązanie twarde tworzymy bez opcji `-s`. Różnice opisuje strona Dowiązania twarde i symboliczne.

ln = link: utwórz dowiązanie do pliku, twarde albo symboliczne z flagą -s

ZMIENIA SYSTEM

Wspólny katalog grupy z dziedziczeniem grupy: chmod 2770

```
sudo chmod 2770 /srv/NazwaKatalogu
```

Bit setgid (cyfra 2 na początku) sprawia, że nowe pliki w katalogu dostają grupę katalogu, a nie grupę twórcy. Trzy pozostałe cyfry to prawa właściciela, grupy i innych.

chmod = change mode: zmień prawa dostępu

ZMIENIA SYSTEM

Zmiana celu, do którego system startuje domyślnie: systemctl set-default

```
sudo systemctl set-default multi-user.target
```

Cel `multi-user.target` to system tekstowy z siecią i usługami, a `graphical.target` dokłada graficzny ekran logowania. Aktualny cel pokazuje `systemctl get-default`.

systemctl = system control: sterowanie usługami systemd

TYLKO ODCZYT

Jednorazowe przejście do innego celu: systemctl isolate

```
sudo systemctl isolate rescue.target
```

Zatrzymuje większość usług i przechodzi do trybu ratunkowego bez restartu. Wracamy komendą `sudo systemctl default`.

systemctl = system control: sterowanie usługami systemd

□□ ZMIENIA SYSTEM

Dodanie parametru do wszystkich jąder: grubby --update-kernel

```
sudo grubby --update-kernel=ALL --args="NazwaParametru=Wartosc"
```

Zmiana ląduje w konfiguracji programu rozruchowego i przetrwa restart. Usunięcie parametru robi opcja `--remove-args`, a podgląd wszystkich wpisów `sudo grubby --info=ALL`.

□□ **grubby** = od GRUB i nazwy narzędzia pomocniczego: zmienia argumenty jądra i domyślny wpis rozruchowy, bez ręcznej edycji plików GRUB

□□ TYLKO ODCZYT

Obniżenie priorytetu działającego procesu: renice

```
sudo renice -n 10 -p NumerPID
```

Dodatnia wartość oznacza łagodniejsze traktowanie procesu przez planistę, a zakres to od minus 20 (najważniejszy) do 19. Zwykły użytkownik może tylko podnosić wartość własnych procesów.

□□ **renice** = od re i nice: zmienia priorytet już działającego procesu, im większa wartość nice, tym proces ustępuje innym

□□ TYLKO ODCZYT

Przełączenie profilu strojenia: tuned-adm profile

```
sudo tuned-adm profile NazwaProfilu
```

Listę dostępnych profili pokazuje `tuned-adm list`, podpowiedź systemu `tuned-adm recommend`, a aktywny profil `tuned-adm active`.

□□ **tuned-adm** = od tuned administration: przełącza profile strojenia systemu usługi tuned, na przykład pod serwer albo pod maszynę wirtualną

□□ TYLKO ODCZYT

Utworzenie tablicy GPT i partycji: parted

```
sudo parted -s /dev/NazwaDysku mklabel gpt mkpart NazwaPartycji 1MiB 1GiB
```

Komenda `mklabel gpt` kasuje poprzednią tablicę partycji, więc używamy jej wyłącznie na pustym dysku. Stan sprawdzamy poleceniem `sudo parted /dev/NazwaDysku print`.

GPT = GUID Partition Table: nowoczesny sposób dzielenia dysku na partycje · **parted** = od partition editor: edytor tablic partycji dla dysków z GPT i MBR, zmiany zapisuje od razu

ZMIENIA SYSTEM

Rozszerzenie wolumenu logicznego razem z systemem plików: lvextend -r

```
sudo lvextend -r -L +500M /dev/NazwaGrupy/NazwaWolumenu
```

Opcja `-r` zmienia też rozmiar systemu plików, więc osobne `xfs_growfs` albo `resize2fs` nie jest potrzebne. System XFS umie tylko rosnąć, nie da się go zmniejszyć.

XFS = system plików stworzony przez Silicon Graphics, domyślny w Rocky Linux i RHEL · **lvextend** = logical volume extend: powiększ wolumen logiczny

TYLKO ODCZYT

Wpis montowania po UUID w pliku fstab

```
UUID=NumerUUID /sklep xfs defaults 0 0
```

UUID odczytujemy komendą `sudo blkid /dev/NazwaGrupy/NazwaWolumenu`. Poprawność wpisu sprawdzamy poleceniem `sudo mount -a`, które montuje wszystko z fstab i zgłasza błąd, zanim system zrobi to przy restarcie.

UUID = Universally Unique Identifier: długi, praktycznie niepowtarzalny identyfikator

TYLKO ODCZYT

Utworzenie i włączenie przestrzeni wymiany: mkswap

```
sudo mkswap /dev/NazwaPartycji
```

Potem `sudo swapon /dev/NazwaPartycji` włącza ją od razu, a wpis `UUID=... none swap defaults 0 0` w fstab na stałe. Stan pokazuje `swapon --show`.

UUID = Universally Unique Identifier: długi, praktycznie niepowtarzalny identyfikator · **mkswap** = od make swap: przygotowuje partycję, wolumen albo plik jako przestrzeń wymiany, zanim włączy

ją swapon

□□ TYLKO ODCZYT

Eksport katalogu przez NFS: exportfs -ra

```
sudo exportfs -ra
```

Wczytuje pliki z `/etc/exports` i `/etc/exports.d/`. Podgląd aktualnego eksportu daje `sudo exportfs -v`.

□□ **NFS** = Network File System: udostępnianie katalogów po sieci w świecie Linuksa ·**exportfs** = od export filesystem: stosuje i wypisuje listę katalogów udostępnianych przez serwer NFS z pliku /etc/exports

□□ TYLKO ODCZYT

Reguła autofs dla katalogu NFS

```
wymiana -fstype=nfs4,rw rhlab02:/srv/nfs/wymiana
```

Wpis trafia do pliku mapy, na przykład `/etc/auto.nfs`, a plik mapy wskazuje drugi plik w `/etc/auto.master.d/`. Katalog montuje się dopiero przy pierwszym dostępie, a po minucie bezczynności odmontowuje.

□□ **NFS** = Network File System: udostępnianie katalogów po sieci w świecie Linuksa

Dzień 1: laboratorium, powłoka, dokumentacja i SSH

□□ Z ŻYCIA WZIĘTE

Wtorek, 9:00. Ola kładzie przed Tomkiem kartkę z jednym zdaniem: „Na egzaminie nikt nie zapyta cię, co robi `-r`, tylko każe zrobić, żeby działało”. Tomek właśnie wpisał w przeglądarkę, jak wyświetlić linie niezakomentowane w pliku, i dostaje od Oli pytanie, czy zrobiłby to samo bez internetu. Odpowiada, że `grep` i `man` ma na każdym serwerze, i zaczyna od `man grep`.

Cele egzaminu

- wpisywanie poleceń o poprawnej składni, przekierowania wejścia i wyjścia
- `grep` i wyrażenia regularne do analizy tekstu
- dostęp do zdalnych systemów przez SSH, logowanie i przełączanie użytkowników w celach wielożytkownikowych
- znajdowanie i czytanie dokumentacji systemowej: `man`, `info`, pliki w `/usr/share/doc`

Czytamy w bazie

- Przekierowania: >, >>, 2>, &>, <, tee
- grep: wyszukiwanie w treści plików i Wyrażenia regularne: BRE, ERE, PCRE
- Jak czytać dokumentację: man, --help, apropos, info, tldr
- SSH: klucze, ssh-agent, config, hartowanie

Ćwiczenia

1. W katalogach `/etc/ssh/sshd_config` i `/etc/ssh/sshd_config.d/` znajdujemy wszystkie aktywne linie, zapisujemy je w pliku `~/ssh-aktywne.txt`, a błędy odrzucamy.

Odpowiedź

```
grep -rEv '^[[:space:]]*(#|$)' /etc/ssh/sshd_config /etc/ssh/sshd_config.d/ > ~/ssh-
aktywne.txt 2>/dev/null
cat ~/ssh-aktywne.txt
```

Przekierowanie `2>/dev/null` wyrzuca komunikaty błędów, a nie wynik. Gdy plik wyjdzie pusty, sprawdzamy, czy nie pomyliliśmy kolejności przekierowań.

1. Bez internetu ustalamy, w której stronie `man` opisany jest format pliku `/etc/fstab`, i otwieramy ją.

Odpowiedź

```
man -k fstab
man 5 fstab
```

Wynik pierwszej komendy zawiera wiersz `fstab (5)`. Jeżeli lista jest pusta, indeks dopiero trzeba zbudować: `sudo mandb`, a potem powtarzamy.

1. Logujemy się z `rhlab01` na `rhlab02`, przechodzimy na roota komendą `su -` i sprawdzamy, czym różni się od zwykłego `su`.

Odpowiedź

```
ssh tomek@rhlab02
su -
pwd
echo "$PATH"
exit
su
pwd
echo "$PATH"
exit
```

Z myślnikiem dostajemy powłokę logowania roota: katalog domowy `/root` i `PATH` roota. Bez myślnika zostajemy w bieżącym katalogu ze środowiskiem dotychczasowego użytkownika. Na egzaminie stosujemy `su -`, bo kolejne polecenia zakładają środowisko roota.

Dzień 2: pliki, archiwa, linki i uprawnienia

> **□□ Z ŻYCIA WZIĘTE**
>
> Czwartek, 15:40. Marek z magazynu pyta, czy przed czyszczeniem katalogu z logami skanerów da się zrobić z niego jeden plik, który zmieści się na pendrive. Tomek pamięta, że `tar` ma kilkanaście opcji, ale nie pamięta, która skompresuje. Zamiast szukać, wpisuje `man tar` i szuka słowa gzip, bo tak będzie musiał robić na egzaminie.

Cele egzaminu

- archiwizowanie i kompresowanie plików: `tar`, `gzip`, `bzip2`

- tworzenie i edycja plików tekstowych
- tworzenie, usuwanie, kopiowanie i przenoszenie plików i katalogów
- twarde i symboliczne dowiązania
- standardowe prawa ugo/rwx

Czytamy w bazie

- System plików, ścieżki, pwd, cd, ls
- Dowiązania twarde i symboliczne
- Uprawnienia klasyczne: chmod, chown, chgrp
- umask, setuid, setgid, sticky bit
- vim: minimum operacyjne dla administratora

Krótko o tym, czego baza nie ma

Baza opisuje kompresję plików z logów, ale nie opisuje samego `tar`. Polecenie ma trzy główne tryby: `-c` tworzy archiwum, `-x` rozpakowuje, `-t` pokazuje zawartość. Kompresję wybiera druga opcja: `-z` to gzip (rozszerzenie `.tar.gz`), `-j` to bzip2 (`.tar.bz2`), `-J` to xz. Opcja `-f` wskazuje plik archiwum i musi stać tuż przed jego nazwą.

Ćwiczenia

1. Robimy skompresowane archiwum katalogu `/etc/ssh` w pliku `/tmp/ssh-kopia.tar.gz`, wypisujemy jego zawartość bez rozpakowania i rozpakowujemy tylko plik `sshd_config` do katalogu `/tmp/odtworzone`.

Odpowiedź

```
sudo tar -czf /tmp/ssh-kopia.tar.gz -C /etc ssh
tar -tzf /tmp/ssh-kopia.tar.gz
mkdir -p /tmp/odtworzone
tar -xzf /tmp/ssh-kopia.tar.gz -C /tmp/odtworzone ssh/sshd_config
ls -l /tmp/odtworzone/ssh/sshd_config
```

Ścieżka w archiwum zaczyna się od `ssh/`, bo `-C /etc` zmienia katalog bazowy. Archiwum w `/tmp` zrobił root, więc do odczytu zwykłemu użytkownikowi wystarczą domyślne prawa (`rw-r--r--`).

1. Tworzymy plik, dowiązanie twarde i dowiązanie symboliczne do niego, a potem kasujemy oryginał. Sprawdzamy, które z dowiązań nadal działa i dlaczego.

Odpowiedź

```
cd ~
echo "Zielona lab" > oryginal.txt
ln oryginal.txt twarde.txt
ln -s oryginal.txt miekkie.txt
ls -li oryginal.txt twarde.txt miekkie.txt
rm oryginal.txt
cat twarde.txt
cat miekkie.txt
```

Dowiązanie twarde ma ten sam numer i-węzła co oryginał, więc plik żyje, dopóki istnieje choć jedna nazwa. Dowiązanie symboliczne zawiera tylko ścieżkę, więc po skasowaniu oryginału `cat` zgłasza brak pliku.

1. Tworzymy grupę `handlowcy` i katalog `/srv/wspolny`: właścicielem ma być root, grupą `handlowcy`, grupa ma pełny dostęp, inni żadnego, a nowe pliki mają dziedziczyć grupę katalogu.

Odpowiedź

```
sudo groupadd handlowcy
sudo mkdir /srv/wspolny
sudo chown root:handlowcy /srv/wspolny
sudo chmod 2770 /srv/wspolny
ls -ld /srv/wspolny
```

Wynik zawiera `drwxrws---`: litera `s` w polu grupy to bit setgid. Test: `sudo touch /srv/wspolny/test` i `ls -l /srv/wspolny` pokazują grupę `handlowcy` także dla pliku utworzonego przez roota.

Dzień 3: proste skrypty powłoki

Z ŻYCIA WZIĘTE

Piątek, 14:20. Tomek sprawdza ręcznie zajętość dysków na czterech serwerach i za każdym razem wpisuje `df -h`, szuka wzrokiem właściwego wiersza i zapomina, co było na poprzednim. Ola mówi, że to jest zadanie na skrypt z jednym parametrem, który ma wypisać jedną linię i skończyć z kodem 1, jeśli jest więcej niż osiemdziesiąt procent. Tomek pisze pierwszą wersję w pięć minut i drugą, poprawioną, w kolejne piętnaście.

Cele egzaminu

- warunkowe wykonywanie kodu: `if`, `test`, `[ ]`
- pętle do przetwarzania plików i danych z linii poleceń
- przetwarzanie parametrów skryptu: `$1`, `$2`
- przetwarzanie wyniku poleceń wewnątrz skryptu

Czytamy w bazie

- Pierwszy skrypt: shebang, uprawnienia, uruchomienie
- Zmienne, parametry i argumenty
- Warunki i pętle
- Kod wyjścia, $?, &&, ||, średnik

Ćwiczenia

1. Piszemy skrypt `/usr/local/bin/sprawdz-dysk.sh`, który przyjmuje punkt montowania jako pierwszy parametr (domyślnie `/`), wypisuje procent zajętości i kończy się kodem 1, gdy zajętość wynosi 80 procent lub więcej.

Odpowiedź

```
#!/bin/bash
punkt="${1:-/}"
proc=$(df --output=pcent "$punkt" | tail -1 | tr -dc '0-9')
if [ "$proc" -ge 80 ]; then
    echo "UWAGA: $punkt zajety w ${proc}%"
    exit 1
fi
```

```
echo "OK: $punkt zajety w ${proc}%"
```

Plik zapisujemy poleceniem `sudo vim /usr/local/bin/sprawdz-dysk.sh`, dajemy mu prawo wykonania (`sudo chmod 755 /usr/local/bin/sprawdz-dysk.sh`) i sprawdzamy kod wyjścia przez `echo $?`. Wartość w nawiasie `${1:-/}` znaczy: pierwszy parametr, a jeśli go nie ma, to `/`.

1. Piszemy skrypt, który czyta `/etc/passwd` i wypisuje nazwę oraz katalog domowy każdego zwykłego użytkownika, czyli takiego z numerem UID od 1000 w górę, z pominięciem konta `nobody`.

Odpowiedź

```
#!/bin/bash
while IFS=: read -r nazwa _ uid _ _ dom _; do
    if [ "$uid" -ge 1000 ] && [ "$nazwa" != "nobody" ]; then
        echo "$nazwa $dom"
    fi
done < /etc/passwd
```

Zmienna `IFS` ustawiona na dwukropek dzieli linię na pola, a znak podkreślenia oznacza pole, którego nie potrzebujemy. Konto `nobody` ma UID 65534, więc bez drugiego warunku też trafiłoby na listę.

1. Piszemy skrypt, który dostaje dowolną liczbę nazw plików i dla każdej wypisuje `istnieje` albo `brak`.

Odpowiedź

```
#!/bin/bash
for plik in "$@"; do
    if [ -e "$plik" ]; then
        echo "$plik: istnieje"
    else
        echo "$plik: brak"
    fi
done
```

Cudzysłowy wokół `"$@"` są ważne, bo bez nich nazwa z odstępem rozpadłaby się na dwa słowa.

Dzień 4: start systemu, cele systemd i hasło roota

Z ŻYCIA WZIĘTE

Wtorek, 16:05. Maszyna testowa po urlopie administratora nie wpuszcza nikogo: hasło roota zna tylko człowiek, który jest na Teneryfie. Pan Wojtek, audytor, uśmiecha się, że dostęp fizyczny do konsoli zawsze był w audytach dziurą, i zostaje na kawie. Tomek restartuje maszynę, naciska `e` w menu programu rozruchowego i po raz pierwszy robi to, czego wymaga egzamin: przerywa rozruch.

Cele egzaminu

- uruchamianie, restart i zatrzymanie systemu
- ręczne uruchamianie systemu w różnych celach (targets)
- przerwanie rozruchu, żeby uzyskać dostęp do systemu
- ustawienie domyślnego celu rozruchu
- zmiana konfiguracji programu rozruchowego

Czytamy w bazie

- systemd i systemctl: jednostki, zależności, usługi
- Kernele: lista, usuwanie starych, brak miejsca na /boot

Krótko o tym, czego baza nie ma

Pojęcie: cel systemd (target). Cel to nazwana grupa usług, do której system ma dojść podczas startu. `multi-user.target` to system tekstowy z siecią, `graphical.target` dokłada ekran graficzny, `rescue.target` to tryb jednego użytkownika z zamontowanymi dyskami, a `emergency.target` to najmniejsza możliwa powłoka z korzeniem w trybie tylko do odczytu.

Pojęcie: program rozruchowy (bootloader). To pierwszy program po firmware, który wybiera jądro i przekazuje mu parametry. W RHEL jest nim GRUB 2, a zmiany parametrów na stałe robi polecenie `grubby`. Jednorazową zmianę, bez zapisywania, robimy w menu GRUB

> klawiszem `e`, a uruchamiamy ją klawiszami Ctrl i x.

Odzyskanie hasła roota na RHEL 9 robiło się parametrem `rd.break`: system zatrzymuje się w powłoce początkowej z korzeniem pod `/sysroot`. Dla RHEL 10 źródła wtórne opisują dwie drogi, `rd.break` i `init=/bin/bash`, a kurs RH134 na RHEL 10 ma rozdział o odzyskiwaniu dostępu administratora. Którą z nich uznajemy za działającą, sprawdzamy sami w ćwiczeniu 3 na maszynie z migawką.

Ćwiczenia

1. Ustawiamy domyślny cel `multi-user.target`, restartujemy, sprawdzamy, że system do niego doszedł, a potem przywracamy poprzedni cel.

Odpowiedź

```
systemctl get-default
sudo systemctl set-default multi-user.target
sudo systemctl reboot
```

Po restarcie:

```
systemctl get-default
systemctl is-active multi-user.target
```

Poprzedni cel, jeżeli był inny, ustawiamy tą samą komendą, na przykład `sudo systemctl set-default graphical.target`.

1. Przed restartem robimy migawkę `przed-dzien4`. Potem jednorazowo uruchamiamy `rhlab02` w trybie `rescue.target`, nie zmieniając konfiguracji na stałe.

Odpowiedź

W menu GRUB zaznaczamy pierwszy wpis i naciskamy `e`. W linii zaczynającej się od `linux` dopisujemy na końcu parametr, a potem startujemy:

```
systemd.unit=rescue.target
Ctrl+x
```

System prosi o hasło roota, bo tryb ratunkowy wymaga logowania. Wyjście z trybu robi `systemctl default` albo zwykły restart, a zmiana w menu nie została zapisana, więc kolejny

rozruch jest normalny.

1. Przywracamy hasło roota na `rhlab02` bez znajomości starego hasła. Zaczynamy od migawki, potem próbujemy `rd.break`, a jeśli się nie uda, `init=/bin/bash`.

Odpowiedź

W menu GRUB naciskamy `e`, w linii `linux` usuwamy `rhgb quiet` i dopisujemy `rd.break`, po czym startujemy klawiszami Ctrl i x. W powłoce początkowej wykonujemy:

```
mount -o remount,rw /sysroot
chroot /sysroot
passwd root
touch /.autorelabel
exit
exit
```

Plik `/.autorelabel` każe SELinuksowi przepisać etykiety plików przy następnym starcie, a bez niego logowanie może się nie udać, bo `/etc/shadow` dostaje złą etykietę. Pierwszy rozruch trwa wtedy dłużej. Drugi sposób: w linii `linux` zmieniamy `ro` na `rw` i dopisujemy `init=/bin/bash`, po starcie wpisujemy `passwd`, `touch /.autorelabel` i `exec /sbin/init`. Który sposób zadziałał na naszym RHEL 10, zapisujemy w dzienniku błędów.

1. Dodajemy do wszystkich jąder parametr `audit=1`, sprawdzamy po restarcie, że jest w `/proc/cmdline`, a potem go usuwamy.

Odpowiedź

```
sudo grubby --update-kernel=ALL --args="audit=1"
sudo grubby --info=ALL | grep args
sudo systemctl reboot
```

Po restarcie:

```
grep audit /proc/cmdline
sudo grubby --update-kernel=ALL --remove-args="audit=1"
```

Zmiana przez `grubby` jest trwała i dotyczy wszystkich zainstalowanych jąder, w odróżnieniu od edycji w menu GRUB, która działa tylko na jeden start.

Dzień 5: procesy, strojenie, dzienniki i przesyłanie plików

> **Z ŻYCIA WZIĘTE**
>
> Poniedziałek, 6:05. W Swarzędzu skanery zalogowały się naraz do systemu magazynowego, a na `ora01` pracuje jeszcze nocna kopia, która od pięciu minut zjada procesor. Marek dzwoni drugi raz, a Tomek, zamiast zabijać proces kopii, obniża mu priorytet i patrzy, jak obciążenie spada. Ola pyta, skąd wiedział, że `kill` w tym przypadku zepsułby kopię, i dopisuje do planu dnia „renice zamiast kill”.

Cele egzaminu

- rozpoznawanie procesów obciążających procesor i pamięć oraz ich zabijanie
- zmiana priorytetu procesów
- zarządzanie profilami strojenia
- znajdowanie i czytanie logów oraz dzienników, zachowywanie dzienników po restarcie
- uruchamianie, zatrzymywanie i sprawdzanie usług sieciowych
- bezpieczne kopiowanie plików między systemami

Czytamy w bazie

- ps: zdjęcie stanu procesów i Monitorowanie na żywo: top, htop, btop
- Sygnały: kill, pkill, killall, trap
- journalctl: logi systemd
- Retencja i czyszczenie dzienników
- Kto nasłuchuje i kto z kim gada: ss, lsof, netstat

Krótko o tym, czego baza nie ma

> **Pojęcie: tuned.** To usługa, która zmienia ustawienia jądra pod rodzaj pracy maszyny. Każdy profil, na przykład `throughput-performance` albo `virtual-guest`, to zestaw takich ustawień. Cel egzaminu mówi o zarządzaniu profilami, więc umiemy sprawdzić aktywny, wskazać zalecany i przełączyć.

Dziennik systemd domyślnie leży w pamięci i znika po restarcie, jeżeli katalog `/var/log/journal` nie istnieje albo ustawienie `Storage` nie jest wymuszone. Dlatego cel „zachowaj dzienniki” oznacza konkretne ustawienie `Storage=persistent`.

Ćwiczenia

1. Uruchamiamy w tle dwa procesy obciążające procesor (`sha256sum /dev/zero &`), znajdujemy je na liście procesów posortowanej według obciążenia procesora, obniżamy priorytet jednego z nich do 15 i kończymy oba.

Odpowiedź

```
sha256sum /dev/zero &
sha256sum /dev/zero &
ps aux --sort=-%cpu | head -5
renice -n 15 -p NumerPID
pkill sha256sum
```

Zwykły użytkownik może obniżyć priorytet własnego procesu bez `sudo`. Gdy zadanie wymaga zmiany priorytetu cudzego procesu albo podniesienia ważności, używamy `sudo`. Komenda `pkill` kończy procesy po nazwie, a bez parametrów wysyła sygnał TERM.

1. Sprawdzamy, jaki profil tuned jest aktywny, który system zaleca, a potem przełączamy na `throughput-performance` i weryfikujemy.

Odpowiedź

```
sudo systemctl enable --now tuned
tuned-adm active
tuned-adm recommend
sudo tuned-adm profile throughput-performance
tuned-adm active
```

Przykład (kształt wyniku, nie zapis z konsoli):

```
Current active profile: throughput-performance
```

Jeżeli pakietu nie ma, doinstalowujemy go przez `sudo dnf install tuned`. Przełączenie jest trwałe, bo profil zapisuje się w `/etc/tuned/active_profile`.

1. Włączamy trwałe zapisywanie dziennika i sprawdzamy po restarcie, że widać poprzedni rozruch.

Odpowiedź

```
sudo mkdir -p /etc/systemd/journald.conf.d
sudo tee /etc/systemd/journald.conf.d/trwaly.conf <<'EOF'
[Journal]
Storage=persistent
EOF
sudo systemctl restart systemd-journald
sudo systemctl reboot
```

Po restarcie:

```
journalctl --list-boots
journalctl -b -1 -p err | tail
```

Lista rozruchów musi mieć co najmniej dwa wiersze. Katalog `/var/log/journal` powstaje sam po restarcie usługi.

1. Kopiujemy katalog `/etc/ssh` z `rhlab01` na `rhlab02` do `/tmp` dwiema metodami: `scp` i `rsync` przez SSH.

Odpowiedź

```
scp -r /etc/ssh tomek@rhlab02:/tmp/ssh-scp
rsync -av -e ssh /etc/ssh/ tomek@rhlab02:/tmp/ssh-rsync/
```

Ukośnik na końcu ścieżki źródłowej w `rsync` znaczy: zawartość katalogu, a bez niego powstałby jeszcze jeden poziom. Część plików w `/etc/ssh` jest nieczytelna dla zwykłego użytkownika (klucze prywatne), więc dla niektórych plików zobaczymy błąd odmowy dostępu.

Dzień 6: magazyn lokalny, partycje GPT i LVM

Z ŻYCIA WZIĘTE

Środa, 7:30. Marek zgłasza, że system magazynowy przestaje przyjmować nowe dostawy, bo wolumen z bazą pod `ora01` zapełnił się do końca. Tomek zna LVM z książki w bazie, ale nigdy nie tworzył go od zera, tylko powiększał. Dopiero w laboratorium, na pustym dysku, rozumie, w jakiej kolejności to wszystko się układa: dysk, partycja, wolumen fizyczny, grupa, wolumen logiczny.

Cele egzaminu

- wyświetlanie, tworzenie i usuwanie partycji na dyskach GPT
- tworzenie i usuwanie wolumenów fizycznych
- przypisywanie wolumenów fizycznych do grup wolumenów
- tworzenie i usuwanie wolumenów logicznych

Czytamy w bazie

- LVM: dysk fizyczny, grupa wolumenów, wolumen logiczny
- Co zajmuje miejsce: df, du, lsblk, lsscsi
- Powiększanie przestrzeni dyskowej bez restartu
- Instalacja Rocky Linux z własnym nazewnictwem VG i LV

Krótko o tym, czego baza nie ma

Pojęcie: GPT. Nowoczesna tablica partycji (GUID Partition Table), która zastąpiła starszą tablicę MBR. Pozwala na dyski większe niż dwa terabajty i ma kopię zapasową tablicy na końcu dysku. Cel egzaminu mówi wprost o dyskach GPT, więc `fdisk`, `gdisk` albo `parted` musimy umieć obsłużyć na GPT.

Baza pokazuje LVM na istniejących dyskach, ale nie opisuje partycjonowania od zera. Używamy `parted` w trybie skryptowym (`-s`), bo działa na jednej linii. W konwencji Zielonej każda grupa wolumenów ma nazwę `vg<Przeznaczenie>`, a wolumen `lv<Przeznaczenie>`, na przykład `vgSklep` i `lvSklep`. Na egzaminie stosujemy dokładnie takie nazwy, jakie podaje zadanie.

Ćwiczenia

1. Na pierwszym dodatkowym dysku maszyny `rhlab01` tworzymy tablicę GPT i dwie partycje: pierwszą o rozmiarze 1 GiB na przestrzeń wymiany, drugą o rozmiarze 2 GiB z flagą LVM.

Odpowiedź

Najpierw migawka `przed-dzien6`, potem:

```
lsblk
sudo parted -s /dev/sdb mklabel gpt mkpart swap linux-swap 1MiB 1GiB mkpart lvm 1GiB 3GiB
set 2 lvm on
sudo parted /dev/sdb print
lsblk /dev/sdb
```

Na kontrolerze VirtIO dysk nazywa się `/dev/vdb`. Początek od 1 MiB zapewnia wyrównanie partycji. Gdy nowe partycje nie pojawiają się w `lsblk`, wymuszamy odczyt tablicy poleceniem `sudo partprobe /dev/sdb`.

1. Z drugiej partycji tworzymy wolumen fizyczny, grupę `vgSklep` z rozmiarem jednostki 8 MiB i wolumen logiczny `lvSklep` o rozmiarze 1 GiB.

Odpowiedź

```
sudo pvcreate /dev/sdb2
sudo vgcreate -s 8M vgSklep /dev/sdb2
sudo lvcreate -L 1G -n lvSklep vgSklep
sudo pvs
sudo vgs
sudo lvs
```

Przykład (kształt wyniku):

```
  LV       VG       Attr       LSize
  lvSklep  vgSklep  -wi-a----- 1.00g
```

Ścieżka urządzenia to `/dev/vgSklep/lvSklep`, a także `/dev/mapper/vgSklep-lvSklep`.

1. Cofamy wszystko: usuwamy wolumen logiczny, grupę, wolumen fizyczny i partycje, żeby zostać z pustym dyskiem.

Odpowiedź

```
sudo lvremove /dev/vgSklep/lvSklep
sudo vgremove vgSklep
sudo pvremove /dev/sdb2
sudo parted -s /dev/sdb rm 2 rm 1
sudo parted /dev/sdb print
```

Kolejność jest odwrotna do tworzenia. Operacje kasujące dane, więc wykonujemy je wyłącznie na dysku laboratoryjnym.

Dzień 7: systemy plików, swap i montowanie przy starcie

> **Z ŻYCIA WZIĘTE**
>
> Czwartek, 8:10. Po nocnych aktualizacjach i restarcie serwera testowego system nie wchodzi do trybu wielu użytkowników, tylko ląduje w trybie awaryjnym z komunikatem o braku dysku. Powodem jest literówka w `fstab`, której nikt nie sprawdził poleceniem `mount -a`. Tomek poprawia wpis w trybie awaryjnym i od tego dnia po każdej zmianie w `fstab` wpisuje `sudo mount -a`, zanim cokolwiek zrestartuje.

Cele egzaminu

- montowanie systemów plików przy starcie po UUID lub etykiecie
- tworzenie, montowanie i używanie systemów plików VFAT, ext4 i XFS
- dodawanie partycji, wolumenów i przestrzeni wymiany bez utraty danych
- rozszerzanie istniejących wolumenów logicznych

Czytamy w bazie

- Montowanie: mount, /etc/fstab, zasoby sieciowe
- Powiększanie przestrzeni dyskowej bez restartu
- SWAP: czym jest, kiedy czyścić, kiedy nie ruszać

Ćwiczenia

1. Odtwarzamy partycje i wolumen `lvSklep` z dnia 6 (partycja 1 GiB na przestrzeń wymiany, partycja 2 GiB z LVM, `vgSklep`, `lvSklep` 1 GiB), zakładamy na wolumenie system XFS i montujemy go na stałe w `/sklep` po UUID.

Odpowiedź

```
sudo parted -s /dev/sdb mklabel gpt mkpart swap linux-swap 1MiB 1GiB mkpart lvm 1GiB 3GiB
set 2 lvm on
sudo pvcreate /dev/sdb2
sudo vgcreate -s 8M vgSklep /dev/sdb2
sudo lvcreate -L 1G -n lvSklep vgSklep
sudo mkfs.xfs /dev/vgSklep/lvSklep
sudo mkdir /sklep
uuid=$(sudo blkid -s UUID -o value /dev/vgSklep/lvSklep)
echo "UUID=${uuid} /sklep xfs defaults 0 0" | sudo tee -a /etc/fstab
sudo systemctl daemon-reload
sudo mount -a
findmnt /sklep
```

Po `mount -a` wolumen musi być zamontowany bez komunikatu o błędzie. Restart sprawdza resztę: po nim `findmnt /sklep` nadal ma pokazać wiersz.

1. Powiększamy wolumen `lvSklep` o 500 MiB razem z systemem plików, nie odmontowując go.

Odpowiedź

```
df -h /sklep
sudo lvextend -r -L +500M /dev/vgSklep/lvSklep
df -h /sklep
```

Opcja `-r` wywołuje `xfs_growfs` dla XFS i `resize2fs` dla ext4. Gdy w grupie zabraknie wolnych jednostek, komenda zgłosi błąd, a wolne miejsce w grupie sprawdza `sudo vgs`.

1. Z pierwszej partycji (1 GiB) robimy przestrzeń wymiany i włączamy ją na stałe, po UUID.

Odpowiedź

```
sudo mkswap /dev/sdb1
sudo swapon /dev/sdb1
uuid=$(sudo blkid -s UUID -o value /dev/sdb1)
echo "UUID=${uuid} none swap defaults 0 0" | sudo tee -a /etc/fstab
sudo systemctl daemon-reload
swapon --show
free -h
```

Po restarcie `swapon --show` ma ponownie pokazać partycję. Bez wpisu w `fstab` przestrzeń wymiany znikałaby z każdym restartem.

1. Na drugim dodatkowym dysku tworzymy dwie partycje po 1 GiB: jedną z systemem VFAT i etykietą `WYMIANA`, drugą z systemem ext4 i etykietą `ARCHIWUM`, i montujemy je na stałe po etykietach w `/wymiana` i `/archiwum`.

Odpowiedź

```
sudo parted -s /dev/sdc mklabel gpt mkpart fat 1MiB 1GiB mkpart ext 1GiB 2GiB
sudo mkfs.vfat -n WYMIANA /dev/sdc1
sudo mkfs.ext4 -L ARCHIWUM /dev/sdc2
sudo mkdir /wymiana /archiwum
echo "LABEL=WYMIANA /wymiana vfat defaults 0 0" | sudo tee -a /etc/fstab
echo "LABEL=ARCHIWUM /archiwum ext4 defaults 0 0" | sudo tee -a /etc/fstab
sudo systemctl daemon-reload
sudo mount -a
findmnt /wymiana /archiwum
```

Do znalezienia etykiet służą `lsblk -f` i `blkid`. Etykietę zmieniamy później na ext4 komendą `e2label`, a na XFS `xfs_admin -L`.

Dzień 8: NFS, autofs i powtórka bloku

Z ŻYCIA WZIĘTE

Środa, 11:00. Ola prosi o katalog wspólny między `app01` i `ci01`, w którym zespół wymienia artefakty budowania, i nie chce kopiować go skryptem co godzinę. Tomek stawia na `rhlab02` eksport NFS i montuje go na `rhlab01`, a potem trafia na ścianę: po restarcie katalog znika, bo zapomniał wpisu w `fstab`. Ola sugeruje autofs, który montuje tylko wtedy, gdy ktoś zajrzy do katalogu.

Cele egzaminu

- montowanie i odmontowywanie udziałów sieciowych NFS
- konfiguracja autofs

Czytamy w bazie

- Montowanie: mount, /etc/fstab, zasoby sieciowe
- firewalld: strefy, porty, usługi, rich rules

Krótko o tym, czego baza nie ma

Pojęcie: NFS. Sieciowy system plików (Network File System): serwer udostępnia katalog, klient montuje go jak lokalny dysk. Serwer w RHEL to usługa `nfs-server` z pakietu `nfs-utils`, a lista eksportów leży w `/etc/exports` i plikach z `/etc/exports.d/`. Klient potrzebuje tylko tego samego pakietu i wpisu w `fstab` z opcją `_netdev`, która mówi: montuj dopiero po uruchomieniu sieci.

Pojęcie: autofs. Usługa, która montuje udział sieciowy dopiero wtedy, gdy ktoś zajrzy do katalogu, i odmontowuje go po czasie bezczynności. Konfiguracja ma dwa pliki: plik główny (mapa nadrzędna) mówi, który katalog jest pod nadzorem, a plik mapy opisuje, co w nim montować.

Ćwiczenia

1. Na `rhlab02` eksportujemy katalog `/srv/nfs/wymiana` dla sieci 10.20.150.0/24 z prawem zapisu i otwieramy zaporę. Na `rhlab01` montujemy go ręcznie w `/mnt/wymiana`.

Odpowiedź

Na `rhlab02`:

```
sudo dnf -y install nfs-utils
sudo mkdir -p /srv/nfs/wymiana
echo '/srv/nfs/wymiana 10.20.150.0/24(rw,sync)' | sudo tee /etc/exports.d/wymiana.exports
sudo systemctl enable --now nfs-server
sudo exportfs -ra
sudo exportfs -v
sudo firewall-cmd --permanent --add-service=nfs
sudo firewall-cmd --reload
```

Na `rhlab01`:

```
sudo dnf -y install nfs-utils
sudo mkdir -p /mnt/wymiana
sudo mount -t nfs rhlab02:/srv/nfs/wymiana /mnt/wymiana
findmnt /mnt/wymiana
sudo touch /mnt/wymiana/test
```

Gdy zapis się nie udaje mimo opcji `rw`, sprawdzamy prawa katalogu na serwerze: eksport nadaje dostęp sieciowy, ale prawa plików nadal obowiązują.

1. Ustawiamy montowanie tego udziału na stałe w `fstab` z opcją `_netdev` i sprawdzamy je poleceniem `mount -a`.

Odpowiedź

```
sudo umount /mnt/wymiana
echo 'rhlab02:/srv/nfs/wymiana /mnt/wymiana nfs _netdev,defaults 0 0' | sudo tee -a
/etc/fstab
sudo systemctl daemon-reload
sudo mount -a
findmnt /mnt/wymiana
```

Opcja `_netdev` zapobiega zawieszeniu rozruchu, gdy sieć nie jest jeszcze gotowa.

1. Zamiast wpisu w `fstab` konfigurujemy autofs: katalog `/mnt/auto` ma montować udział `wymiana` dopiero przy pierwszym dostępie i odmontowywać po 60 sekundach.

Odpowiedź

Najpierw usuwamy wpis NFS z `fstab` i odmontowujemy udział. Potem:

```
sudo dnf -y install autofs
echo '/mnt/auto /etc/auto.nfs --timeout=60' | sudo tee /etc/auto.master.d/nfs.autofs
echo 'wymiana -fstype=nfs4,rw rhlab02:/srv/nfs/wymiana' | sudo tee /etc/auto.nfs
sudo systemctl enable --now autofs
ls /mnt/auto/wymiana
findmnt /mnt/auto/wymiana
```

Katalog `/mnt/auto/wymiana` nie istnieje do pierwszego dostępu, więc `ls` na nim wywołuje montowanie. Odmontowanie po minucie bezczynności sprawdzamy poleceniem `findmnt` po krótkiej przerwie. Dla katalogów domowych używamy wzorca `* -fstype=nfs4,rw rhlab02:/srv/nfs/home/&`, w którym gwiazdka i znak `&` oznaczają tę samą nazwę.

1. Mini sprawdzian na czas, trzydzieści minut, bez notatek: tworzymy grupę `serwis`, katalog `/srv/serwis` z prawami 2770 i grupą `serwis`, wolumen logiczny `lvSerwis` w grupie `vgSerwis` na dysku `sdc` (1 GiB, XFS), zamontowany na stałe w `/srv/serwis` po UUID, plus skrypt `/usr/local/bin/ile-wolnego.sh`, który wypisuje wolne miejsce w `/srv/serwis` w megabajtach. Po restarcie wszystko musi działać.

Odpowiedź

```
sudo groupadd serwis
sudo parted -s /dev/sdc mklabel gpt mkpart lvm 1MiB 3GiB set 1 lvm on
sudo pvcreate /dev/sdc1
sudo vgcreate vgSerwis /dev/sdc1
sudo lvcreate -L 1G -n lvSerwis vgSerwis
sudo mkfs.xfs /dev/vgSerwis/lvSerwis
sudo mkdir -p /srv/serwis
uuid=$(sudo blkid -s UUID -o value /dev/vgSerwis/lvSerwis)
echo "UUID=${uuid} /srv/serwis xfs defaults 0 0" | sudo tee -a /etc/fstab
sudo systemctl daemon-reload
sudo mount -a
sudo chown root:serwis /srv/serwis
sudo chmod 2770 /srv/serwis
```

Skrypt:

```
#!/bin/bash
df -BM --output=avail /srv/serwis | tail -1 | tr -dc '0-9'
echo " MB wolnego"
```

Dysk `sdc` po ćwiczeniu z dnia 7 ma tablicę GPT z dwiema partycjami, więc przed sprawdzianem wracamy do migawki `czysty-start` albo czyścimy go komendą `sudo wipefs -a /dev/sdc`. Uwaga: ta komenda niszczy dane na dysku laboratoryjnym.

Częste błędy

Objaw	Przyczyna	Co robimy
Po restarcie dysk nie jest zamontowany	wpis w `fstab` pominięty albo z literówką	zawsze `sudo mount -a` po edycji, potem restart
System startuje do trybu awaryjnego po zmianie `fstab`	błędny wpis w pliku	w trybie awaryjnym `mount -o remount,rw /`, poprawiamy plik
`lvextend` zgłasza brak miejsca	grupa nie ma wolnych jednostek	`sudo vgs`, dokładamy wolumen fizyczny przez `vgextend`
`parted` zgłasza, że partycja się nie mieści	złe jednostki albo początek 0	podajemy `MiB` i początek od `1MiB`
Nowa partycja nie widać w `lsblk`	jądro nie odczytało tablicy	`sudo partprobe NazwaDysku`
Montowanie NFS zawiesza się	zapora zamyka port 2049 albo brak sieci	`sudo firewall-cmd --add-service=nfs` na serwerze, test `ping rhlab02`
Po resecie hasła roota nie można się zalogować	pominięty plik `/.autorelabel`	powtarzamy procedurę z dopisaniem pliku i czekamy na relabel
`man -k` nic nie znajduje	brak indeksu stron podręcznika	`sudo mandb`, potem powtarzamy

Co dalej

Po tygodniu pierwszym wracamy do dziennika błędów i powtarzamy z pamięci to, co nie wyszło. Kolejne dni opisuje strona Tydzień 2 do EX200: oprogramowanie, czas, sieć, użytkownicy i bezpieczeństwo.

Źródła: strona Red Hat o egzaminie EX200 (cele, stan na 10 października 2026); strona Red Hat o kursie RH134 na RHEL 10 (zakres: odzyskiwanie dostępu administratora, LVM, NFS); computingforgeeks.com, opis dwóch metod resetu hasła roota na RHEL 10; Red Hat, Considerations in adopting RHEL 10 (zmiany w sieci i magazynie).

Revision #8
Created 2026-10-10 13:47:32 UTC by Tomasz Konieczny
Updated 2026-10-26 14:25:17 UTC by Tomasz Konieczny