Instalace a software

Modul 02

Sergey Kuroedov 2026

Cíle a obsah modulu

1. **Způsoby instalace** — zvolit způsob instalace a ověřit původ instalačního image
2. **Průběh instalace** — navrhnout rozdělení disku s ohledem na pozdější změny
3. **Práce v příkazovém řádku** — spravovat soubory a archivy, vyhledat text a upravit soubor v editoru
4. **Správa balíčků** — nainstalovat, aktualizovat a dohledat balíček nástrojem `dnf`
5. **Repozitáře** — připojit repozitář třetí strany a omezit jeho rozsah

2.1 Způsoby instalace

Instalační image, Kickstart, cloud image

Sergey Kuroedov 2026

Dostupnost distribuce

- Fedora Server stáhnete bez registrace z oficiálních mirrorů
- image **netinst** zavede instalátor a balíčky stáhne ze sítě
- image **dvd** obsahuje balíčky pro instalaci bez sítě
- pro virtualizaci a cloud slouží hotové image `qcow2` a AMI
- u každého image ověřte checksum i podpis GPG

```
curl -O https://fedoraproject.org/fedora.gpg
gpgv --keyring ./fedora.gpg <soubor_checksum>
sha256sum -c --ignore-missing <soubor_checksum>
```

Způsoby instalace

Způsob	Typické použití
interaktivní instalace z ISO image	jednotlivé stroje, zkušební nasazení
Kickstart	hromadné a opakovatelné nasazení
síťová instalace PXE	nasazení bez fyzického média
cloud image	virtualizace a cloud
container image	runtime prostředí aplikace

Instalace nástrojem Kickstart

- soubor Kickstart obsahuje odpovědi na všechny otázky instalátoru
- instalátor zapíše po každé instalaci soubor `/root/anaconda-ks.cfg`
- cestu k souboru předáte parametrem jádra `inst.ks=`
- syntaxi ověříte nástrojem `ksvalidator` ještě před nasazením

```
ksvalidator /root/anaconda-ks.cfg
```

Osvědčený postup — Vyjděte ze souboru `anaconda-ks.cfg` z referenční instalace. Psát Kickstart od začátku je zdrojem chyb.

2.2 Průběh instalace

Anaconda, profily softwaru, rozdělení disku

Sergey Kuroedov 2026

Instalační program Anaconda

- Anaconda nabízí grafický i textový režim, volitelně přístup přes VNC
- rozdělení disku provedete automaticky nebo ručně
- profil softwaru určuje výchozí sadu nainstalovaných balíčků
- v instalátoru nastavíte síť, časovou zónu a uživatelské účty
- logy instalace najdete po dokončení v `/var/log/anaconda/`

Profily softwaru

- **Fedora Server Edition** — výchozí serverová sada s nástrojem Cockpit
- **Minimal Install** — základní systém s nejmenší attack surface
- **Fedora Custom Operating System** — jádro a správce balíčků, nic dalšího

Doplňky typu Windows File Server zvolíte vedle profilu. Zvolený profil doplníte po instalaci skupinou balíčků, viz oddíl 2.4.

Nastavení s dlouhodobým dopadem

- změna rozdělení disku vyžaduje odstávku, případně novou instalaci
- LVM umožní pozdější zvětšení bez přeinstalace, viz modul 06
- oddíly `/var` a `/home` lze při instalaci oddělit od systémového
- souborový systém XFS je ve Fedora Server výchozí a nelze jej zmenšit
- SELinux zapnutý až po instalaci vyžaduje relabel celého systému

> **Upozornění** — Rozdělení disku navrhujte s rezervou. Logický svazek LVM zvětšíte snadno, souborový systém XFS nezmenšíte vůbec.

2.3 Práce v příkazovém řádku

Shell, soubory, archivy, text, roury a editory

Sergey Kuroedov 2026

Prompt shellu

```
[jnovak@web01 ~]$ sudo -i
[root@web01 ~]# exit
[jnovak@web01 ~]$
```

- prompt uvádí účet, název stroje a aktuální adresář
- znak `$` označuje běžný účet, znak `#` účet root
- klávesa `Tab` doplní název příkazu, souboru nebo adresáře
- šipka nahoru a `Ctrl+R` vyhledají příkaz v historii
- `Ctrl+C` přeruší běžící příkaz

Stavba příkazu

```
ls -l -a --human-readable /var/log
│  └─┘ └──────────────────┘ └─────┘
│    │                     └─ argument: s čím příkaz pracuje
│    └─ krátké přepínače
└─ příkaz
```

- krátký přepínač má jednu pomlčku, dlouhý dvě
- krátké přepínače lze sloučit: `ls -l -a` je totéž co `ls -la`
- `<prikaz> --help` vypíše stručnou nápovědu
- `man <prikaz>` zobrazí úplnou dokumentaci příkazu

Adresářový strom

Adresář	Obsah
`/`	kořenový adresář, z něhož vychází celý strom
`/home/<ucet>`	domovské adresáře uživatelů
`/root`	domovský adresář účtu root
`/etc`	konfigurace systému a služeb
`/var/log`	logy systému a aplikací

> **Poznámka** — Strom nepoužívá písmena jednotek. Další disky se připojují do adresářů, podrobně v modulu 06.

Absolutní a relativní cesta

```
$ pwd
/home/jnovak
$ cd /etc/ssh
$ cd ../yum.repos.d
$ pwd
/etc/yum.repos.d
```

- absolutní cesta začíná znakem `/` a platí odkudkoli
- relativní cesta vychází z aktuálního adresáře
- `.` označuje aktuální adresář, `..` nadřazený adresář
- `~` označuje domovský adresář přihlášeného účtu

Pohyb v adresářích

Příkaz	Účel
`pwd`	vypíše cestu k aktuálnímu adresáři
`cd <adresar>`	přejde do zadaného adresáře
`cd ..`	přejde do nadřazeného adresáře
`cd`	přejde do domovského adresáře
`cd -`	vrátí se do předchozího adresáře

Výpis obsahu adresáře

Příkaz	Účel
`ls -l`	vypíše oprávnění, vlastníka, velikost a čas změny
`ls -a`	zahrne skryté soubory, jejichž název začíná tečkou
`ls -lh`	uvede velikost v jednotkách K, M a G
`ls -ltr`	seřadí podle času změny, nejnovější na konci
`ls -ld <adresar>`	vypíše údaje o adresáři, nikoli o jeho obsahu

Dlouhý výpis

```
$ ls -lh /etc/hosts
-rw-r--r--. 1 root root 158 Jan 24  2025 /etc/hosts
$ ls -ld /var/log
drwxr-xr-x. 12 root root 4.0K Nov  8 09:12 /var/log
```

- první znak určuje typ: `-` soubor, `d` adresář, `l` symbolický odkaz
- dalších devět znaků tvoří přístupová oprávnění, podrobně v modulu 09
- následuje počet odkazů, vlastník, skupina, velikost a čas změny
- tečka za oprávněními označuje kontext SELinux

Vytvoření adresářů a souborů

```
mkdir zalohy
mkdir -p projekt/data/2026
mkdir -pv web/{dev,test,prod}
touch poznamky.txt
rmdir zalohy
```

- `-p` vytvoří i chybějící nadřazené adresáře, bez chyby při existenci
- `-v` vypíše každý vytvořený adresář
- `touch` vytvoří prázdný soubor nebo aktualizuje čas změny
- `rmdir` odstraní pouze prázdný adresář

Kopírování, přesun a odstranění

Příkaz	Účel
`cp <zdroj> <cil>`	zkopíruje soubor
`cp -r <adresar> <cil>`	zkopíruje adresář včetně obsahu
`cp -a <zdroj> <cil>`	zkopíruje a zachová oprávnění, vlastníka a časy
`mv <zdroj> <cil>`	přesune nebo přejmenuje soubor či adresář
`rm -r <adresar>`	odstraní adresář včetně obsahu

Upozornění — `rm` nepoužívá koš a odstranění nelze vrátit.
Přepínač `-i` vyžádá potvrzení každého souboru.

Zobrazení obsahu souboru

Příkaz	Účel
`cat <soubor>`	vypíše celý soubor najednou
`less <soubor>`	zobrazí soubor po stránkách, klávesa `q` ukončí
`head -n 20 <soubor>`	vypíše prvních 20 řádků
`tail -n 20 <soubor>`	vypíše posledních 20 řádků
`tail -f <soubor>`	průběžně vypisuje nově přidané řádky

Archiv `tar`

```
tar -czvf zaloha-etc.tar.gz /etc
tar -tzf zaloha-etc.tar.gz
tar -xzf zaloha-etc.tar.gz -C /tmp/obnova
tar -cJf logy.tar.xz /var/log/httpd
```

- `c` vytvoří archiv, `t` vypíše obsah, `x` rozbalí
- za `f` následuje název archivu, proto stojí ve skupině poslední
- `z` komprimuje nástrojem gzip, `J` nástrojem xz
- `-C` určí existující adresář, do kterého se archiv rozbalí

Archiv `zip` a komprese `gzip`

```
zip -r projekt.zip projekt/
unzip -l projekt.zip
unzip projekt.zip -d /tmp/obnova
gzip velky.log
gunzip velky.log.gz
```

- `zip` slouží k výměně dat se systémem Windows
- `zip` neuchová vlastníka a skupinu souborů
- `gzip` komprimuje jediný soubor a původní soubor nahradí
- zálohy a přenos mezi linuxovými stroji provádějte nástrojem `tar`

Cvičení 2.1 — Práce se soubory a archivy

Zadání

1. Zjistěte, ve kterém adresáři se po přihlášení nacházíte.
2. Vytvořte jedním příkazem adresáře `~/projekt/data/2026`.
3. Zkopírujte do `~/projekt/data` adresář `/etc/yum.repos.d`.
4. Vypište `~/projekt` rekurzivně se skrytými soubory a čitelnou velikostí.
5. Zabalte `~/projekt` do archivu `projekt.tar.gz` a vypište jeho obsah.
6. Rozbalte archiv do `/tmp/obnova` a porovnejte jej s originálem.

Očekávaný výsledek — Rozbalený adresář je totožný s originálem.
Dokážete vysvětlit, na jaké cestě se v `/tmp/obnova` objevil.

Čas — 15 minut

Výpis textu a přesměrování

```
echo "Přihlášen je $USER"
echo 'Přihlášen je $USER'
echo "první řádek" > poznamky.txt
echo "druhý řádek" >> poznamky.txt
ls /etc /neexistuje 2> chyby.txt
ls /etc /neexistuje > vse.txt 2>&1
```

- `echo` vypíše zadaný text na standardní výstup
- `>` výstup zapíše do souboru a soubor přepíše, `>>` připojí na konec
- `2>` přesměruje chybový výstup, `2>&1` jej spojí se standardním
- ve dvojitých uvozovkách shell dosadí proměnné, v jednoduchých ne

Roura

```
ls -l /etc | less
dnf list --installed | wc -l
sort /etc/passwd | head -n 5
sudo journalctl -b | grep -i error | tail -n 20
```

- znak `|` předá výstup příkazu na vstup následujícího příkazu
- každý příkaz dělá jednu věc, roura je spojí do celku
- rourou prochází pouze standardní výstup, chyby nikoli

Vyhledávání textu příkazem `grep`

Příkaz	Výsledek
`grep root /etc/passwd`	řádky, které obsahují `root`
`grep -i error app.log`	totéž bez ohledu na velikost písmen
`grep -v '^#' sshd_config`	řádky, které vzor neobsahují
`grep -n Listen httpd.conf`	výsledek s čísly řádků
`grep -r PermitRootLogin /etc/ssh`	hledání ve všech souborech adresáře

Textové editory na serveru

```
rpm -q nano vim-minimal vim-enhanced
sudo dnf install -y nano vim-enhanced
echo $EDITOR
```

- `nano` — jednoduchý editor, ovládání vypisuje na spodním řádku
- `vim` — výkonný editor, dostupný na každém unixovém systému
- Fedora má jako výchozí editor nastavený `nano`
- plný `vim` je v balíčku `vim-enhanced`, `vi` v `vim-minimal`

Editor `nano`

Klávesa	Akce
`Ctrl+O`, `Enter`	uloží soubor
`Ctrl+X`	ukončí editor, při neuložených změnách se zeptá
`Ctrl+W`	vyhledá text
`Ctrl+K` / `Ctrl+U`	vyjme / vloží řádek
`Alt+U`	vrátí poslední změnu

Režimy editoru `vim`

```
normální režim  ──  i, a, o  ──▶  vkládací režim
normální režim  ◀──   Esc    ──   vkládací režim
normální režim  ──    :      ──▶  příkazový řádek
```

- po spuštění je `vim` v normálním režimu, klávesy jsou příkazy
- text se píše pouze ve vkládacím režimu
- `Esc` vždy vrátí editor do normálního režimu
- `:wq` uloží a ukončí, `:q!` ukončí bez uložení

Základní příkazy `vim`

Příkaz	Akce
`:w` / `:wq` / `:q!`	uloží / uloží a ukončí / ukončí bez uložení
`dd` / `yy` / `p`	smaže / zkopíruje / vloží řádek
`u` / `Ctrl+R`	vrátí / zopakuje změnu
`/text`, `n`	vyhledá text, přejde na další výskyt
`gg` / `G` / `:25`	přejde na začátek / konec / řádek 25

Cvičení 2.2 — Text, roury a editory

Zadání

1. Příkazem `echo` vytvořte soubor `~/poznamky.txt` se dvěma řádky.
2. Zjistěte, kolik účtů v `/etc/passwd` má shell `/bin/bash`.
3. Vypište aktivní řádky `/etc/ssh/sshd_config` bez komentářů a prázdných řádků.
4. Ověřte, zda je nainstalován `vim`, a případně jej doinstalujte.
5. V editoru `nano` přidejte do `~/poznamky.txt` třetí řádek.
6. V editoru `vim` smažte první řádek a soubor uložte.

Očekávaný výsledek — `cat ~/poznamky.txt` vypíše dva řádky.

Čas — 15 minut

2.4 Správa balíčků

Nástroje `rpm` a `dnf`, aktualizace, historie transakcí

Sergey Kuroedov 2026

Dvě úrovně správy balíčků

- `rpm` pracuje s jedním balíčkem a závislosti neřeší
- `dnf` ve verzi 5 řeší závislosti a stahuje balíčky z repozitářů
- `yum` zůstává zachován jako symbolický odkaz na `dnf`
- databázi nainstalovaných balíčků vede systém v `/var/lib/rpm`

> **Poznámka** — Pro běžnou správu používejte `dnf`. Nástroj `rpm` má smysl pro dotazy do databáze a pro jednotlivý soubor `.rpm`.

Základní operace dnf

Příkaz	Účel
`dnf search <vyraz>`	hledá v názvech a popisech balíčků
`dnf info <balicek>`	vypíše verzi, velikost a repozitář
`dnf install <balicek>`	nainstaluje balíček včetně závislostí
`dnf remove <balicek>`	odstraní balíček a nepotřebné závislosti
`dnf upgrade`	aktualizuje všechny balíčky systému

Dotazy na nainstalovaný stav

Příkaz	Účel
`dnf list --installed`	vypíše nainstalované balíčky
`dnf provides */<nazev>`	najde balíček dodávající daný soubor
`rpm -qa`	vypíše balíčky z databáze bez přístupu k síti
`rpm -qf <cesta>`	určí balíček, kterému soubor patří
`rpm -ql <balicek>`	vypíše soubory dodané balíčkem

Aktualizace systému

- `dnf check-upgrade` vypíše dostupné aktualizace bez zásahu do systému
- `dnf upgrade --security` omezí aktualizaci na bezpečnostní errata
- `dnf needs-restarting -r` posoudí, zda je nutný restart systému
- `dnf system-upgrade` provede přechod na novější release Fedory
- aktualizace jádra ponechá předchozí verzi v nabídce bootloaderu

> **Upozornění** — Fedora má podporu přibližně třináct měsíců. Přechod na novější release je plánovaná činnost, nikoli volitelná.

Historie a vrácení změn

- `dnf history list` vypíše číslované transakce včetně data a rozsahu
- `dnf history info <id>` zobrazí obsah konkrétní transakce
- `dnf history undo <id>` vrátí transakci zpět
- vrácení neobnoví konfigurační soubory změněné po instalaci

Skupiny balíčků

- `dnf group list` vypíše dostupné skupiny balíčků
- `dnf group info <skupina>` vypíše balíčky, které skupina obsahuje
- `dnf group install "Development Tools"` nainstaluje celou skupinu
- skupiny odpovídají profilům softwaru z instalačního programu

Rodina Debian — ekvivalenty

Úloha	RHEL — `dnf`	Debian — `apt`
hledání balíčku	`dnf search`	`apt search`
instalace	`dnf install`	`apt install`
obnovení metadat	provádí automaticky	`apt update`
aktualizace systému	`dnf upgrade`	`apt upgrade`
dotaz na vlastníka souboru	`rpm -qf`	`dpkg -S`

2.5 Repozitáře

Konfigurace, podpisy, repozitáře třetích stran

Sergey Kuroedov 2026

Repozitář

- repozitář tvoří adresář s balíčky a sadou metadat
- metadata popisují balíčky, jejich závislosti a bezpečnostní errata
- konfiguraci repozitářů najdete v `/etc/yum.repos.d/*.repo`
- `dnf` ověřuje podpis GPG u metadat i u každého balíčku

Konfigurace repozitáře

```
[interni]
name=Interní repozitář organizace
baseurl=https://repo.example.com/fedora/$releasever/$basearch/
enabled=1
gpgcheck=1
gpgkey=https://repo.example.com/RPM-GPG-KEY-interni
```

- `enabled=1` zapne repozitář, `gpgcheck=1` vynutí ověření podpisu
- `baseurl` ukazuje na adresář s metadaty, `metalink` na seznam mirrorů

Správa repozitářů

Příkaz	Účel
`dnf repolist`	vypíše povolené repozitáře
`dnf repolist --all`	vypíše i zakázané repozitáře
`dnf config-manager setopt <id>.enabled=1`	povolí repozitář
`dnf config-manager addrepo --from-repofile=<url>`	přidá repozitář
`dnf clean all`	vyčistí uloženou cache metadat

Rizika repozitářů třetích stran

- repozitář třetí strany může nahradit balíček z distribuce
- služba COPR hostuje repozitáře jednotlivců bez kontroly Fedory
- parametr `includepkgs` omezí repozitář na vyjmenované balíčky
- parametr `priority` určí, který repozitář má přednost
- `createrepo_c` vytvoří metadata nad adresářem s vlastními balíčky

DEMONSTRACE

Instalace balíčku a rozbor závislostí

```
dnf repoquery --requires httpd
dnf install httpd
dnf info httpd
rpm -ql httpd | head -20
```

- `repoquery` vypíše závislosti bez jakéhokoli zásahu do systému
- `dnf info` u nainstalovaného balíčku uvede repozitář původu
- `rpm -ql` vypíše soubory, které balíček do systému přidal

Cvičení 2.3 — Správa balíčků a repozitářů

Zadání

1. Zjistěte, který balíček dodává příkaz `ss`.
2. Nainstalujte balíček `httpd` a vypište soubory, které do systému přidal.
3. Určete, ze kterého repozitáře balíček pochází.
4. Vypište poslední transakci `dnf` a zjistěte, co obsahovala.
5. Vraťte systém do stavu před instalací pomocí historie transakcí.

Očekávaný výsledek — Balíček `httpd` není nainstalován a výpis `dnf history list` obsahuje transakci instalace i jejího vrácení.

Čas — 20 minut

Shrnutí modulu

Shrňte vlastními slovy, jak ve své praxi:

- připravíte instalaci serveru a ověříte instalační image
- najdete, upravíte a zazálohujete soubory v shellu
- nainstalujete, aktualizujete a dohledáte balíček
- připojíte repozitář třetí strany bez rizika pro systém

Odkazy na dokumentaci

- Dokumentace Fedory — https://docs.fedoraproject.org
- Fedora Server — https://docs.fedoraproject.org/en-US/fedora-server/
- Příručka nástroje DNF 5 — https://dnf5.readthedocs.io
- Syntaxe Kickstart — https://pykickstart.readthedocs.io
- Služba COPR — https://copr.fedorainfracloud.org
- Rozložení adresářového stromu — `man hier`