Linux — administrace systému

Modul 01 — Úvod

Sergey Kuroedov 2026

Cíle a obsah modulu

1. **Vznik a historie systému** — zařadit Linux mezi systémy odvozené od Unixu a určit verzi jádra
2. **Principy a architektura** — vysvětlit rozdělení systému na jádro a user space
3. **Filozofie vývoje a licence** — rozlišit copyleftové a permisivní licence a jejich dopad
4. **Distribuce** — zvolit distribuci podle modelu releasů a délky podpory

1.1 Vznik a historie systému

Předchůdci systému, vznik jádra, projekt GNU

Sergey Kuroedov 2026

Vznik a vývoj Unixu

- Unix vznikl roku 1969 v Bell Labs; autory jsou Thompson a Ritchie
- přepsáním do jazyka C roku 1973 systém získal přenositelnost
- vývoj se následně rozdělil do dvou linií, System V a BSD
- norma POSIX z roku 1988 standardizovala rozhraní mezi aplikací a systémem

> **Poznámka** — Linux nepřebírá zdrojový kód Unixu. Shoduje se s ním v principech a rozhraní; implementaci má však vlastní.

Vznik jádra Linux

- Linus Torvalds oznámil projekt roku 1991 ve skupině `comp.os.minix`
- cílem bylo volně dostupné jádro pro architekturu i386
- roku 1992 autor převedl jádro pod licenci GPL verze 2
- verze 1.0 z roku 1994 umožnila první produkční nasazení
- dnes jádro tvoří základ serverových systémů i platformy Android

Projekt GNU a doplnění systému

- Richard Stallman zahájil roku 1983 projekt svobodného systému
- do roku 1991 projekt dokončil nástroje a kompilátor, chybělo však jádro
- nástroje GNU doplněné jádrem Linux tvoří teprve úplný systém
- z tohoto spojení vychází označení GNU/Linux

> **Poznámka** — Termín *Linux* označuje v odborném kontextu pouze jádro.
> V běžné praxi se jím rozumí celá distribuce.

Vývoj jádra dnes

- vývoj koordinuje Linus Torvalds, přispívají stovky organizací
- nový release vychází přibližně každých devět až deset týdnů
- vybrané releasy nesou označení LTS a mají delší dobu údržby
- vendoři distribucí dodávají vlastní build s vybranými patchi

> **Upozornění** — Verze jádra v podnikové distribuci neodpovídá releasu z `kernel.org`. Vendor backportuje patche bez změny čísla verze.

Číslování verzí jádra

- schéma má tvar `<major>.<minor>.<patch>`, například `6.12.9`
- zvýšení čísla major neznamená zpětnou nekompatibilitu
- patch release obsahuje pouze opravy chyb a zabezpečení
- příznak `-rc` označuje release candidate, nikoli produkční verzi
- build distribuce doplňuje vlastní příznak

```
$ uname -r
6.11.4-301.fc41.x86_64
```

1.2 Principy a architektura

Vrstvy systému a konstrukční zásady

Sergey Kuroedov 2026

Vrstvy systému

- **jádro** — správa procesů, paměti, zařízení a souborových systémů
- **systémová volání** — jediné rozhraní mezi aplikací a jádrem
- **systémová knihovna** — `glibc` zpřístupňuje volání programovacímu jazyku
- **user space** — nástroje, služby, uživatelské aplikace

Oddělení kernel space a user space zajišťuje, že chyba aplikace neohrozí stabilitu jádra.

Konstrukční zásady

- jeden nástroj řeší jedinou úlohu a řeší ji dobře
- textový výstup slouží jako univerzální rozhraní mezi nástroji
- systém ukládá konfiguraci do textových souborů, nikoli do databáze
- zásada „vše je soubor“ platí pro zařízení, procesy i síťová spojení
- systém odděluje mechanismus od politiky jeho využití

Vše je soubor — důsledky

- zařízení jsou přístupná v `/dev` běžnými nástroji pro práci se soubory
- stav jádra je čitelný z `/proc` a `/sys`
- na zařízení platí stejný model oprávnění jako na běžný soubor

```
$ grep -m1 "model name" /proc/cpuinfo
model name      : Intel(R) Xeon(R) Gold 6248R CPU @ 3.00GHz
```

Víceuživatelský provoz

- systém počítá od počátku se souběžnou prací více účtů
- každý proces má vlastníka a z něj odvozená oprávnění
- u účtu `root` systém přístupová oprávnění nekontroluje
- oddělení účtů tvoří základ bezpečnostního modelu systému

> **Osvědčený postup** — Pracujte pod běžným účtem a oprávnění roota získávejte příkazem `sudo` pro konkrétní úlohu.

Rozdíly proti prostředí Windows

Oblast	Windows	Linux
Konfigurace	registry	textové soubory v `/etc`
Oddělovač cesty	`\`	`/`
Velikost písmen v názvech	nerozlišuje	rozlišuje
Instalace software	instalátor u každé aplikace	repozitář a balíčkovací systém
Otevřený soubor	zámek brání přejmenování	přejmenování i odstranění možné

1.3 Filozofie vývoje a licence

Svobodný software, copyleft, dopady na provoz

Sergey Kuroedov 2026

Svobodný a otevřený software

- licence svobodného software zaručuje čtyři svobody: spouštění, studium, šíření a úpravu programu
- *free software* zdůrazňuje svobodu uživatele
- *open source* zdůrazňuje vývojový model a dostupnost kódu
- množiny licencí se v praxi z velké části překrývají
- dostupnost kódu neznamená absenci licenčních podmínek

Typy licencí

Typ	Příklady	Odvozené dílo
Copyleft	GPL v2, GPL v3	musí zachovat stejnou licenci
Slabý copyleft	LGPL, MPL	lze použít i v uzavřené aplikaci
Permisivní	MIT, BSD, Apache 2.0	lze šířit pod vlastní licencí

- jádro Linux podléhá licenci GPL verze 2
- zavedení proprietárního modulu označí jádro příznakem *taint*

Dopady na provoz

- distribuci lze provozovat bez poplatku za licenci software
- u komerčních distribucí hradí subscription podporu a certifikace
- software se poskytuje bez záruky; záruku poskytuje vendor podpory
- interní úprava bez šíření nezakládá povinnost zveřejnit kód
- šíření upraveného díla pod GPL vyžaduje zpřístupnění zdrojového kódu

> **Upozornění** — Zahrnutí komponenty pod GPL do produktu určeného zákazníkům má licenční důsledky. Postup konzultujte s právním útvarem.

1.4 Distribuce

Rodiny, modely releasů, životní cyklus

Sergey Kuroedov 2026

Distribuce

Vendor distribuce integruje, ověřuje a dodává:

- jádro v konkrétní verzi s vlastními patchi
- sadu ověřených balíčků a nástroj pro jejich správu
- instalační program a výchozí konfiguraci systému
- definovaný životní cyklus a kanál pro bezpečnostní aktualizace
- dokumentaci a případně komerční podporu

Hlavní rodiny distribucí

Rodina	Správa balíčků	Představitelé
Red Hat	`rpm`, `dnf`	RHEL, Rocky Linux, AlmaLinux, Fedora
Debian	`dpkg`, `apt`	Debian, Ubuntu
SUSE	`rpm`, `zypper`	SLES, openSUSE
Nezávislé	vlastní nástroje	Arch Linux, Alpine Linux, Gentoo

Modely releasů

- **stabilní release** — pevný obsah, dlouhý životní cyklus (RHEL, Debian, SLES)
- **rolling release** — průběžná aktualizace balíčků (Arch, Tumbleweed)
- **upstream** — zdroj pro komerční distribuci (Fedora pro RHEL)

Volba je kompromisem mezi stabilitou prostředí a dostupností nových verzí.

Životní cyklus a výběr distribuce

- podnikové distribuce poskytují podporu deset let i více
- volbu určuje podpora vendora hardwaru, podpora aplikací a certifikace
- rebuildy nabízejí binární kompatibilitu bez subscription
- jednotná distribuce v organizaci snižuje náklady na správu

> **Osvědčený postup** — Před nasazením ověřte, do kdy vendor dodává bezpečnostní aktualizace, a naplánujte migraci před koncem tohoto období.

Referenční prostředí kurzu

Všechna cvičení probíhají v tomto prostředí:

- distribuce **Fedora Server** — upstream RHEL, rodina Red Hat
- správce balíčků `dnf` ve verzi 5
- jádro z aktuální řady, bez backportů vendora
- dva stroje pro každého účastníka, přístup přes SSH pod účtem `uzivatel`
- oprávnění roota příkazem `sudo`

> **Poznámka** — Odchylky podnikových distribucí a rodiny Debian uvádíme u jednotlivých témat, kde se liší.

Přístup k cvičným strojům

```
ssh amenit@sergeykuroedov.cz
ssh uzivatel@192.168.122.159
```

Stanoviště	Stroj a	Stroj b
0	[DOPLNIT]	192.168.122.96
1	192.168.122.159	192.168.122.146
2	192.168.122.173	192.168.122.33
3	192.168.122.8	192.168.122.252

Účet `uzivatel`, heslo `ABCabc123`

DEMONSTRACE

Identifikace systému

```
cat /etc/os-release
uname -srm
hostnamectl
```

- `/etc/os-release` — název a verze distribuce ve strojově čitelném tvaru
- `uname` — verze jádra a architektura
- `hostnamectl` — souhrnné údaje včetně názvu stroje

Cvičení 1.1 — Identifikace prostředí

Zadání

1. Připojte se přes přístupový server ke stroji a svého stanoviště.
2. Zjistěte název a verzi distribuce bez použití příkazu `cat`.
3. Zjistěte verzi jádra a architekturu procesoru.
4. Určete, kdy byl systém naposledy spuštěn.
5. Zapište, zda je jádro označeno příznakem *taint*.

Očekávaný výsledek — Pět zjištěných údajů zaznamenaných v poznámkách, u každého název použitého příkazu.

Čas — 10 minut

Shrnutí modulu

Shrňte vlastními slovy, jak ve své praxi:

- vysvětlíte kolegovi z Windows, čím se Linux liší
- zjistíte, jakou distribuci a jádro server používá
- posoudíte licenční dopad převzetí open source komponenty
- vyberete distribuci pro nový produkční server

Odkazy na dokumentaci

- Zdrojové kódy a releasy jádra — https://www.kernel.org
- Licence GNU a jejich výklad — https://www.gnu.org/licenses
- Schválené licence open source — https://opensource.org/licenses
- Dokumentace Fedory — https://docs.fedoraproject.org
- Manuálové stránky online — https://man7.org/linux/man-pages