Obsah

Počítačové sítě

1. Referenční model ISO/OSI

Popis vrstev, protokol, PDU, enkapsulace

1.1. Úvod do referenčního modelu ISO/OSI

Referenční model ISO/OSI (Open Systems Interconnection) je standardizovaný model pro síťovou komunikaci, který byl vyvinut Mezinárodní organizací pro normalizaci (ISO). Tento model rozděluje komunikaci mezi počítači do sedmi vrstev, kde každá vrstva má své specifické funkce a protokoly. Cílem je zajistit interoperabilitu mezi různými systémy a zařízeními v síti.

1.2. Popis vrstev referenčního modelu OSI

Referenční model OSI se skládá ze sedmi vrstev:
1. Fyzická vrstva (Physical Layer)

- **Účel:** Stará se o fyzický přenos datových signálů mezi síťovými zařízeními.
- **Funkce:** Definuje typ média (např. kabely, optická vlákna), konektory, elektrické signály a přenosovou rychlost. Řeší konverzi binárních dat do signálů, které jsou přenášeny po fyzickém médiu.
- **Příklady zařízení:** Huby, kabely, opakovače.
- **PDU:** Bit

2. Linková vrstva (Data Link Layer)

- **Účel:** Zajišťuje přenos dat mezi dvěma sousedními síťovými uzly.
- **Funkce:** Detekuje a opravuje chyby při přenosu, ověřuje integritu dat a organizuje je do rámců (frame). Obsahuje podvrstvu LLC (Logical Link Control) a MAC (Media Access Control).
- **Příklady protokolů:** Ethernet, PPP, Wi-Fi (802.11).
- **PDU:** Rámec (Frame)

3. Síťová vrstva (Network Layer)

- **Účel:** Řídí směrování dat mezi zařízeními v různých sítích, poskytuje adresování a přenos dat mezi sítěmi.
- **Funkce:** Směruje datové pakety na správnou adresu (IP adresa), rozhoduje o nejlepších trasách a provádí fragmentaci a de-fragmentaci paketů.
- **Příklady protokolů:** IP (IPv4, IPv6), ICMP.
- **PDU:** Paket (Packet)

4. Transportní vrstva (Transport Layer)

- **Účel:** Zajišťuje spolehlivý přenos dat mezi hostiteli v síti, zajišťuje správné pořadí dat a jejich úplnost.
- **Funkce:** Segmentuje data do menších částí, zajišťuje kontrolu přenosu (kontrola chyb, toků) a zajišťuje doručení správnému aplikačnímu procesu.
- **Příklady protokolů:** TCP (spolehlivý přenos), UDP (nespolehlivý přenos).
- **PDU:** Segment (Segment) pro TCP a Datagram pro UDP

5. Relační vrstva (Session Layer)

- **Účel:** Řídí a udržuje komunikaci mezi dvěma aplikacemi (např. synchronizace dialogů).
- **Funkce:** Umožňuje vytváření, udržování a ukončení relace mezi aplikacemi. Řídí synchronizaci a správu datových toků.
- Příklady protokolů: NetBIOS, RPC.
- **PDU:** Data

6. Prezentační vrstva (Presentation Layer)

- **Účel:** Starost o formátování a prezentaci dat, která jsou zasílána do aplikační vrstvy.
- **Funkce:** Konverze dat mezi formáty (např. ASCII, JPEG), šifrování a dešifrování dat, komprese a dekomprese.
- **Příklady:** SSL/TLS pro šifrování, MPEG, JPEG pro formátování.
- **PDU:** Data

7. Aplikační vrstva (Application Layer)

- **Účel:** Poskytuje síťové služby pro aplikace, které uživatelé přímo používají.
- **Funkce:** Nabízí síťové služby, které umožňují interakci s aplikacemi, jako jsou e-mailové programy, webové prohlížeče a vzdálená správa.
- **Příklady protokolů:** HTTP, FTP, SMTP, DNS.
- **PDU:** Data

1.3. Protokol

Protokol je soubor pravidel a formátů, které určují, jak mají zařízení komunikovat. Každá vrstva referenčního modelu OSI má vlastní protokoly, které definují, jak mají být data zpracována a přenášena.
Příklady protokolů v jednotlivých vrstvách:

- Fyzická vrstva: USB, Ethernet
- Linková vrstva: Ethernet, ARP
- Síťová vrstva: IP, ICMP
- Transportní vrstva: TCP, UDP
- Aplikační vrstva: HTTP, FTP, DNS

1.4. PDU (Protocol Data Unit)

Protokolová datová jednotka (PDU) je formát dat na jednotlivých vrstvách modelu OSI:

- Fyzická vrstva – **Bity**
- Linková vrstva – **Rámec (Frame)**
- Síťová vrstva – **Paket (Packet)**
- Transportní vrstva – Segment (Segment) nebo Datagram
- Vyšší vrstvy (Relační, Prezentační, Aplikační) – **Data**

Každá vrstva přidává k datům své informace (hlavičku a někdy i patičku), což vytváří novou PDU. Tento proces je důležitý pro směrování a interpretaci dat při jejich cestě sítí.

1.5. Enkapsulace

Enkapsulace je proces, při kterém každá vrstva přidává k datům svou hlavičku (a někdy i patičku) s potřebnými informacemi pro správné doručení a zpracování dat v síti. Tento proces začíná na aplikační vrstvě a končí na fyzické vrstvě, kde jsou data přenesena ve formě signálů.
Proces enkapsulace zahrnuje:

1. **Aplikační vrstva** generuje data pro komunikaci.
2. **Prezentační vrstva** může data šifrovat a komprimovat.
3. **Relační vrstva** přidává informace pro synchronizaci.
4. **Transportní vrstva** rozděluje data na segmenty, přidává hlavičku s informacemi o číslech portů.
5. **Síťová vrstva** přidává IP hlavičku, která obsahuje cílovou a zdrojovou IP adresu.
6. **Linková vrstva** přidává MAC adresy a vytváří rámec (Frame).
7. **Fyzická vrstva** převádí rámec na bity a posílá ho po fyzickém médiu.

1.6. Závěr

Referenční model ISO/OSI je důležitým konceptem pro pochopení síťové komunikace. Každá vrstva má jasně definované úkoly a protokoly, které zajišťují interoperabilitu mezi různými zařízeními. PDU a enkapsulace jsou klíčové procesy pro efektivní přenos dat sítí. Tento model slouží jako základní struktura, na které jsou založeny moderní síťové technologie a standardy.

2. Technologie Ethernet

Přenosová média, metoda přístupu, druhy přenosu

2.1. Úvod do technologie Ethernet

Ethernet je dominantní technologie pro lokální sítě (LAN – Local Area Network), která umožňuje vzájemnou komunikaci mezi zařízeními, jako jsou počítače, servery a routery. Byl vyvinut v 70. letech minulého století a postupně se stal standardem pro přenos dat v rámci sítí LAN. Ethernet definuje pravidla pro fyzická přenosová média, metodu přístupu k síti a různé druhy přenosu dat.
Ethernet je standardizován v normě **IEEE 802.3**, která zahrnuje různé varianty a rychlosti, od 10 Mb/s až po 400 Gb/s.

2.2. Přenosová média v Ethernetu

Ethernet podporuje různé typy přenosových médií, která se liší vlastnostmi a určením podle vzdálenosti, přenosové rychlosti a použití. Nejčastěji se používají následující média:
a) Kroucená dvoulinka (Twisted Pair)

- **Kategorie:** UTP (Unshielded Twisted Pair), STP (Shielded Twisted Pair)
- **Klasifikace dle kategorie:** Kategorie 5e, 6, 6a, 7, 8 (čím vyšší číslo, tím vyšší rychlost a nižší rušení).
- **Použití:** Nejčastější médium pro Ethernet v běžných LAN, například kategorie 5e a 6 podporují rychlost až 1 Gb/s, kategorie 6a a 7 umožňují přenosy až 10 Gb/s.
- **Dosah:** Typicky až 100 metrů.
- **Výhody:** Nízká cena, flexibilita, snadná instalace.
- **Nevýhody:** Omezený dosah a citlivost na elektromagnetické rušení (snížená u STP kabelů).

b) Optická vlákna (Fiber Optic)

- **Typy:** Single-mode (jednomódové) a Multi-mode (vícemódové) optické vlákno.
- **Použití:** Používá se v páteřních spojích a pro dlouhé vzdálenosti v rámci datových center, kampusových sítí nebo metropolitních sítí.
- **Rychlosti:** Podporuje velmi vysoké rychlosti, včetně 10 Gb/s, 40 Gb/s, 100 Gb/s a více.
- **Dosah:** Multi-mode – stovky metrů; Single-mode – až několik desítek kilometrů.
- **Výhody:** Odolnost proti elektromagnetickému rušení, vysoké rychlosti a dlouhý dosah.
- **Nevýhody:** Vyšší cena a náročnější instalace.

c) Koaxiální kabel

- **Typy:** Tenký koaxiální kabel (10BASE2) a silný koaxiální kabel (10BASE5).
- **Použití:** Původně běžně používaný v Ethernet sítích, dnes je již téměř nahrazen kroucenou dvojlinkou a optickými vlákny.
- **Dosah:** U 10BASE2 až 185 metrů, u 10BASE5 až 500 metrů.
- **Výhody:** Relativně odolný proti rušení.
- **Nevýhody:** Omezená rychlost (pouze 10 Mb/s), obtížnější manipulace, téměř nepoužívaný v moderních sítích.

2.3. Metoda přístupu v Ethernetu

Ethernet využívá metodu přístupu **CSMA/CD (Carrier Sense Multiple Access with Collision Detection)**. Tato metoda určuje, jak se zařízení v síti chovají při pokusu o vysílání dat:
CSMA/CD

- **Carrier Sense (Detekce nosné):** Každé zařízení „poslouchá“ médium (kabel) a čeká, dokud není volné. Pokud médium obsazeno není, zařízení může zahájit přenos.
- **Multiple Access (Vícenásobný přístup):** Více zařízení sdílí jedno médium a mohou k němu přistupovat současně.
- **Collision Detection (Detekce kolize):** Pokud dojde ke kolizi (současné vysílání dvou zařízení), je přenos přerušen a zařízení čekají náhodně dlouhou dobu, než se pokusí přenos zopakovat. Tento postup zajišťuje, že kolize se budou opakovat stále méně často.

Poznámka: Metoda CSMA/CD je využívána především v původních Ethernet sítích s huby a společným přenosovým médiem. V moderních Ethernet sítích se především používají switche (přepínače), které umožňují plně duplexní provoz (přenos dat oběma směry současně) a eliminují kolize, takže CSMA/CD se již běžně neuplatňuje.

2.4. Druhy přenosu v Ethernetu

Ethernet podporuje různé druhy přenosu v závislosti na použitých zařízeních a přenosových režimech:
a) Poloduplexní přenos (Half-Duplex)

- **Charakteristika:** Přenos dat probíhá střídavě jedním směrem, tzn. zařízení mohou buď vysílat, nebo přijímat, ale ne současně.
- **Využití:** Používá se v sítích s huby nebo v sítích, kde není nutný simultánní přenos dat.
- **Nevýhody:** Dochází ke kolizím, což vyžaduje použití CSMA/CD pro řešení kolizí.

b) Plný duplexní přenos (Full-Duplex)

- **Charakteristika:** Umožňuje současný přenos dat v obou směrech, tedy zařízení mohou vysílat i přijímat současně.
- **Využití:** Standardní přenosový režim u moderních Ethernet sítí s přepínači (switch), kde nedochází ke kolizím.
- **Výhody:** Zvyšuje efektivitu sítě, zajišťuje vyšší přenosové rychlosti, eliminuje potřebu CSMA/CD.

c) Unicast, Broadcast a Multicast přenosy

- **Unicast:** Přenos od jednoho zařízení k přesně jednomu dalšímu zařízení. Nejčastější typ komunikace, kde data jsou doručena konkrétnímu cílovému zařízení (např. přenos mezi dvěma počítači).
- **Broadcast:** Přenos od jednoho zařízení všem zařízením v síti (např. ARP dotaz pro zjištění MAC adresy).
- **Multicast:** Přenos od jednoho zařízení skupině zařízení, které se přihlásily k odběru specifického datového toku (např. streamování videa do více zařízení najednou).

2.5. Rychlosti Ethernetu

Ethernet nabízí různé standardy rychlostí přenosu:

- **10BASE-T:** 10 Mb/s (Ethernet)
- **100BASE-TX:** 100 Mb/s (Fast Ethernet)
- **1000BASE-T:** 1 Gb/s (Gigabit Ethernet)
- **10GBASE-T:** 10 Gb/s (10 Gigabit Ethernet)
- 40GBASE-T: 40 Gb/s
- 100GBASE-X: 100 Gb/s
- 400GBASE-X: 400 Gb/s

Každý standard má své požadavky na médium, dosah a další specifikace. Například vyšší rychlosti jako 10GBASE-T a více obvykle vyžadují optická vlákna nebo vysokokvalitní kategorie kroucené dvoulinky (např. Cat 6a, Cat 7).

2.6. Závěr

Ethernet je široce používaná technologie pro LAN sítě, která definuje standardy pro přenosová média, metody přístupu a různé druhy přenosů. Podporuje jak poloduplexní, tak plně duplexní přenosy a umožňuje unicast, multicast a broadcast komunikaci. Rychlosti Ethernetu se neustále zvyšují, což umožňuje efektivní přenos dat i v moderních sítích s vysokými nároky na šířku pásma.

3. Bezdrátové sítě

Metoda přístupu, zabezpečení, základní struktura bezdrátové sítě

3.1. Úvod do bezdrátových sítí

Bezdrátové sítě umožňují komunikaci mezi zařízeními bez použití kabelů, což poskytuje flexibilitu a mobilitu. Nejčastějším standardem pro bezdrátové sítě je **Wi-Fi** (Wireless Fidelity), který je definován specifikací **IEEE 802.11**. Tento standard pokrývá různé verze s odlišnými přenosovými rychlostmi, šířkami kanálů a frekvencemi.
Bezdrátové sítě se využívají v různých prostředích, od domácností po podnikové sítě, a umožňují snadné připojení zařízení, jako jsou notebooky, telefony a IoT zařízení, k síti a internetu.

3.2. Metoda přístupu v bezdrátových sítích

Bezdrátové sítě používají metodu přístupu **CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)**. Tato metoda je nezbytná pro prevenci kolizí v bezdrátovém prostředí, kde zařízení nemohou detekovat kolize stejně jako u kabelových sítí.
CSMA/CA – Přístupová metoda v bezdrátových sítích

- **Carrier Sense (Detekce nosné):** Zařízení „poslouchá“ médium (vzduch) a čeká, dokud je médium volné. Pokud médium není obsazeno, zařízení může zahájit přenos.
- **Collision Avoidance (Prevence kolizí):** Před zahájením přenosu zařízení odešle krátký signál (RTS – Request to Send) a počká na potvrzení (CTS – Clear to Send) od přístupového bodu (Access Point, AP). Tento mechanismus umožňuje zařízení „vyhradit“ si médium pro svůj přenos, čímž se snižuje pravděpodobnost kolizí.
- **Backoff Mechanism:** V případě, že médium není dostupné, zařízení čeká náhodně dlouhou dobu (backoff čas) a poté se pokusí znovu.

Poznámka: Na rozdíl od CSMA/CD, který se používá u kabelových sítí, metoda CSMA/CA nedetekuje kolize, ale předchází jim, což je důležité v prostředí, kde není možné fyzicky detekovat kolizi mezi signály různých zařízení.

3.3. Zabezpečení bezdrátových sítí

Zabezpečení bezdrátových sítí je klíčové, protože signály mohou být zachyceny kýmkoli v dosahu. Existuje několik metod zabezpečení, které chrání bezdrátovou síť před neoprávněným přístupem a zajišťují důvěrnost dat:
a) WEP (Wired Equivalent Privacy)

- **Popis:** První bezpečnostní protokol pro Wi-Fi sítě, nyní zastaralý.
- **Šifrování:** Používá statický 40bitový nebo 104bitový klíč pro šifrování.
- **Nevýhody:** Snadno prolomitelný kvůli slabé implementaci šifrování a opakujícím se inicializačním vektorům. V současnosti se nedoporučuje používat.

b) WPA (Wi-Fi Protected Access)

- **Popis:** Zlepšení WEP, které přidává dynamické klíče a lepší šifrovací mechanismus.
- **Šifrování:** TKIP (Temporal Key Integrity Protocol), který mění šifrovací klíče pro každou datovou jednotku.
- **Nevýhody:** Stále relativně zranitelný, přestože je bezpečnější než WEP.

c) WPA2 (Wi-Fi Protected Access 2)

- **Popis:** Nahrazuje WPA a je výrazně bezpečnější, široce používaný v moderních sítích.
- **Šifrování:** Používá AES (Advanced Encryption Standard), což je robustní šifrovací algoritmus, který poskytuje vysokou úroveň zabezpečení.
- **Nevýhody:** Vysoká úroveň zabezpečení, ale zranitelné vůči některým útokům, jako je KRACK (Key Reinstallation Attack), pokud není aktualizován.

d) WPA3

- **Popis:** Nejnovější bezpečnostní standard pro Wi-Fi sítě, navržený pro vyšší bezpečnost i v osobních sítích.
- **Šifrování:** Používá Simultaneous Authentication of Equals (SAE), což je protokol odolný proti útokům typu offline dictionary.
- **Výhody:** Zvyšuje bezpečnost pomocí silnějšího šifrování a chrání sítě s otevřeným přístupem prostřednictvím funkce OWE (Opportunistic Wireless Encryption), která šifruje komunikaci i bez hesla.

e) Další metody zabezpečení

- **Skrytí SSID:** Skrytí identifikátoru sítě (SSID), což znemožňuje její snadné nalezení. Je to ale jen základní ochrana, kterou lze obejít.
- **Filtrace MAC adres:** Omezení přístupu na základě MAC adres zařízení, ale lze obejít klonováním MAC adresy.
- **VPN (Virtual Private Network):** Doporučeno v nezabezpečených sítích pro vytvoření bezpečného šifrovaného spojení.

3.4. Základní struktura bezdrátové sítě

Bezdrátové sítě mohou být organizovány různými způsoby v závislosti na jejich účelu a prostředí. Mezi hlavní typy patří:
a) Infrastrukturní režim (Infrastructure Mode)

- **Popis:** Tradiční typ bezdrátové sítě, ve které jsou všechna zařízení připojena k centrálnímu přístupovému bodu (AP).
- **Využití:** Používá se v domácnostech, kancelářích a veřejných Wi-Fi sítích, kde AP poskytuje přístup k síti i internetu.
- **Výhody:** Přístupový bod spravuje veškerou komunikaci, zařízení nemohou komunikovat přímo mezi sebou, což zvyšuje bezpečnost.
- **Příklad:** Domácí Wi-Fi síť, podniková Wi-Fi síť.

b) Ad hoc režim

- **Popis:** V tomto režimu zařízení komunikují přímo mezi sebou bez potřeby přístupového bodu.
- **Využití:** Vhodné pro dočasné sítě, jako jsou konference, schůzky nebo síťování v terénu, kde není k dispozici infrastruktura.
- **Výhody:** Rychlá a snadná konfigurace, nezávislost na AP.
- **Nevýhody:** Menší dosah, nižší stabilita a bezpečnost ve srovnání s infrastrukturním režimem.

c) Režim Mesh (Mesh Network)

- **Popis:** Zařízení v síti Mesh spolupracují a každé zařízení (uzel) může být jak AP, tak klient. Vytváří se zde tzv. „překrývající“ síť, kde každé zařízení může komunikovat s jinými uzly, čímž se dosahuje vyššího pokrytí a stability.
- **Využití:** Většinou v rozsáhlých prostředích, jako jsou velké podnikové nebo venkovní sítě, kde je nutné pokrýt velkou oblast.
- **Výhody:** Vysoká odolnost proti výpadkům (pokud jeden uzel selže, data mohou jít přes jiný uzel), velký dosah pokrytí.
- **Nevýhody:** Složitější správa, vyšší náklady.

3.5. Typy bezdrátových sítí dle rozsahu

Bezdrátové sítě lze také rozlišit podle dosahu a určení:

- **PAN (Personal Area Network):** Osobní síť s malým dosahem, například Bluetooth připojení mezi mobilním telefonem a sluchátky.
- **LAN (Local Area Network):** Síť s dosahem typicky do 100 metrů, používající Wi-Fi v domácnostech, kancelářích nebo kavárnách.
- **MAN (Metropolitan Area Network):** Bezdrátová síť pokrývající městskou oblast, například WiMAX.
- **WAN (Wide Area Network):** Síť s velkým dosahem, často zahrnující připojení mezi městy nebo státy, například mobilní sítě 4G/5G.

3.6. Frekvenční pásma v bezdrátových sítích

Bezdrátové sítě mohou používat různá frekvenční pásma, která určují dosah a přenosovou rychlost:

- **2,4 GHz:** Dlouhý dosah, menší přenosová rychlost, náchylnost k rušení (využívají jej např. mikrovlnné trouby, Bluetooth).
- **5 GHz:** Vyšší přenosová rychlost, menší dosah, méně rušení.
- **6 GHz:** Novější pásmo, podporované standardem Wi-Fi 6E, nabízí vysoké rychlosti a méně přetížené kanály.

3.7. Závěr

Bezdrátové sítě poskytují uživatelům velkou flexibilitu a pohodlí, ale vyžadují pečlivé plánování a zabezpečení. Metoda přístupu CSMA/CA umožňuje minimalizaci kolizí, zatímco různé zabezpečovací metody (WPA3, AES) zajišťují ochranu před neoprávněným přístupem. Struktura bezdrátových sítí může být nastavena jako infrastrukturní, ad hoc nebo mesh, což umožňuje jejich nasazení v různých prostředích od domácích až po rozsáhlé podnikové sítě.

4. Aktivní síťové prvky

Rozdělení, parametry, konfigurace

4.1. Úvod do aktivních síťových prvků

Aktivní síťové prvky jsou zařízení, která umožňují přenos dat mezi různými částmi sítě a zajišťují komunikaci mezi připojenými zařízeními. Aktivní prvky pracují s elektrickou nebo optickou energií k posílení, směrování, přeposílání nebo úpravě signálů, což zajišťuje funkčnost a efektivitu síťové komunikace.
Mezi hlavní aktivní síťové prvky patří **huby, switche, routery, access pointy (přístupové body)** a **firewally**. Každý z těchto prvků má svou specifickou funkci a uplatnění v síti.

4.2. Rozdělení aktivních síťových prvků

Aktivní síťové prvky lze rozdělit podle jejich funkce a role v síti:
a) Hub (Rozbočovač)

- **Funkce:** Přenáší data do všech portů kromě toho, ze kterého data přišla. Huby pracují na 1. vrstvě OSI modelu (fyzická vrstva) a nemají schopnost filtrovat nebo směrovat data.
- **Použití:** V moderních sítích jsou již zastaralé a nahrazené switchem, dříve se používaly k propojení malého počtu zařízení.

b) Switch (Přepínač)

- **Funkce:** Přepíná datové pakety mezi zařízeními v síti a rozhoduje, kam je přeposlat, na základě MAC adres. Switch pracuje na 2. vrstvě OSI modelu (linková vrstva) a umožňuje přímou komunikaci mezi zařízeními, čímž zvyšuje efektivitu sítě.
- **Typy:** Existují unmanaged switche (bez možnosti konfigurace) a managed switche (s možností konfigurace).
- **Použití:** Používají se ve většině moderních sítí, a to jak v domácích, tak i podnikových.

c) Router (Směrovač)

- **Funkce:** Směruje datové pakety mezi různými sítěmi, pracuje na 3. vrstvě OSI modelu (síťová vrstva). Routery umožňují komunikaci mezi LAN a internetem a podporují různé protokoly pro směrování.
- **Použití:** Jsou nezbytné pro propojení různých sítí a pro připojení k internetu.

d) Access Point (Přístupový bod)

- **Funkce:** Poskytuje bezdrátové připojení k síti pro Wi-Fi zařízení a rozšiřuje síťovou dostupnost na místa, kde není možné kabelové připojení.
- **Použití:** Používá se v bezdrátových sítích (WLAN) jako přístupový bod pro mobilní zařízení a notebooky.

e) Firewall

- **Funkce:** Zajišťuje bezpečnost sítě tím, že kontroluje a omezuje příchozí a odchozí síťový provoz na základě definovaných pravidel. Firewally mohou být softwarové nebo hardwarové.
- **Použití:** Chrání sítě před neoprávněným přístupem a útoky, používají se ve firemních sítích i v domácnostech pro zabezpečení připojení k internetu.

4.3. Parametry aktivních síťových prvků

Každý aktivní síťový prvek má specifické parametry, které určují jeho výkon, kapacitu a schopnosti v rámci sítě:
a) Hub

- **Počet portů:** Obvykle 4 až 24 portů, ale porty nejsou inteligentně řízené.
- **Rychlost přenosu:** Obvykle 10/100 Mb/s.
- **Možnost konfigurace:** Huby obvykle nejsou konfigurovatelné.

b) Switch

- **Počet portů:** Od 4 do více než 48 portů.
- **Rychlost přenosu:** 10/100/1000 Mb/s nebo až 10 Gb/s na port.
- **Podpora VLAN:** Managed switche umožňují rozdělit síť na více virtuálních LAN (VLAN) pro zvýšení bezpečnosti a efektivity.
- **Podpora PoE (Power over Ethernet):** Možnost napájet připojená zařízení přes Ethernet kabel.

- **Možnost konfigurace:** Managed switche umožňují konfiguraci, např. nastavení VLAN, QoS (Quality of Service) a sledování provozu.

c) Router

- **Počet portů:** LAN a WAN porty pro propojení interní sítě s internetem.
- **Rychlost přenosu:** Záleží na typu routeru, domácí routery mají běžně rychlost kolem 1 Gb/s, profesionální routery zvládají i vyšší přenosy.
- **Podpora NAT (Network Address Translation):** Překládá privátní IP adresy na veřejnou IP adresu, což umožňuje bezpečné připojení k internetu.
- **Podpora VPN (Virtual Private Network):** Umožňuje bezpečné připojení k síti přes veřejné internetové připojení.
- **Možnost konfigurace:** Možnost nastavení směrování, firewallu, QoS a dalších funkcí.

d) Access Point

- **Rychlost přenosu:** Závisí na standardu Wi-Fi (např. 802.11n, 802.11ac, 802.11ax), může být až několik Gb/s.
- **Frekvence:** Podpora pásma 2,4 GHz, 5 GHz a nově 6 GHz pro Wi-Fi 6E.
- **Bezpečnostní protokoly:** WPA2, WPA3 pro zabezpečení bezdrátové komunikace.
- **Možnost konfigurace:** Možnost nastavení SSID, bezpečnostních protokolů, kanálů a šířky pásma.

e) Firewall

- **Průchodnost (Throughput):** Maximální objem dat, který může firewall zpracovat (např. 1 Gb/s, 10 Gb/s).
- **Počet souběžných připojení:** Počet připojení, která firewall zvládne obsloužit současně.
- **Možnosti filtrování:** Filtrace na základě IP adres, portů, protokolů, aplikací.
- **Možnost konfigurace:** Nastavení pravidel pro povolení nebo zakázání konkrétního provozu, filtrování URL adres, detekce útoků (IDS/IPS).

4.4. Konfigurace aktivních síťových prvků

a) Konfigurace switchů

- **Unmanaged switch:** Nevyžaduje žádnou konfiguraci, zařízení stačí připojit a funguje automaticky.
- Managed switch:
    - **Přístup:** Konfigurovatelné přes CLI (Command Line Interface), webové rozhraní nebo SNMP (Simple Network Management Protocol).
    - **Nastavení VLAN:** Oddělení sítě na jednotlivé segmenty pro zvýšení bezpečnosti a správy provozu.
    - **QoS (Quality of Service):** Nastavení priorit pro různé typy provozu (např. priorita pro hlasovou komunikaci).
    - **Spanning Tree Protocol (STP):** Zamezení smyčkám v síti, které by mohly způsobit zahlcení.
    - **Monitoring:** Sledování provozu a vytížení portů, možnost nastavení alertů.

b) Konfigurace routerů

- **Přístup:** Konfigurace přes webové rozhraní, CLI nebo SNMP.
- **Nastavení směrování:** Definice statických nebo dynamických směrovacích cest (pomocí protokolů jako OSPF, RIP, BGP).
- **NAT (Network Address Translation):** Nastavení překladu IP adres pro komunikaci mezi interní sítí a internetem.
- **Firewall:** Konfigurace pravidel pro filtrování provozu, blokování neautorizovaného přístupu.
- **VPN:** Nastavení VPN tunelů pro bezpečné připojení k síti z internetu.

c) Konfigurace access pointů

- **SSID a zabezpečení:** Nastavení názvu sítě (SSID) a bezpečnostních protokolů (WPA2, WPA3) pro ochranu sítě.
- **Kanál a šířka pásma:** Výběr kanálu a šířky pásma (20 MHz, 40 MHz, 80 MHz) pro optimalizaci výkonu a snížení rušení.
- **Frekvenční pásmo:** Výběr pásma (2,4 GHz, 5 GHz, 6 GHz) v závislosti na síťovém zatížení a dosahu.
- **QoS:** Nastavení priorit pro různé typy provozu (např. pro VoIP).

d) Konfigurace firewallů

- **Pravidla pro přístup:** Nastavení pravidel pro povolení nebo zablokování určitého síťového provozu.
- **Filtrování:** Filtrování provozu na základě IP adres, portů a protokolů.
- **VPN:** Konfigurace VPN pro bezpečné propojení mezi lokalitami nebo pro vzdálený přístup.
- **Monitorování a logování:** Sledování provozu, záznam podezřelých aktivit a generování reportů.

- **IDS/IPS:** Detekce a prevence útoků pro ochranu proti malwaru a jiným síťovým hrozbám.

4.5. Závěr

Aktivní síťové prvky jsou klíčovými komponentami každé sítě. Jsou nezbytné pro správné fungování a správu datového provozu v síti, ať už se jedná o domácí nebo podnikové sítě. Huby, switche, routery, access pointy a firewally mají specifické funkce, které podporují efektivní a bezpečnou komunikaci. Správná konfigurace a znalost parametrů aktivních prvků jsou nezbytné pro optimalizaci sítě a zabezpečení datového provozu.

5. IP adresa

IPv4, IPv6, analýza IP adresy, DHCP

5.1. Úvod do IP adres

IP (Internet Protocol) adresa je jedinečný identifikátor zařízení v síti, který umožňuje jejich vzájemnou komunikaci. IP adresy se používají k určení zdrojové a cílové adresy při přenosu dat mezi zařízeními. Existují dvě hlavní verze IP adres: **IPv4** a **IPv6**.

5.2. IPv4 adresa

IPv4 je nejstarší verze IP adresování, která se používá od 80. let. IPv4 adresy jsou široce rozšířené, ale kvůli omezenému množství dostupných adres je IPv4 postupně nahrazováno IPv6.

Struktura IPv4 adresy

- IPv4 adresa je **32bitová** adresa, což znamená, že obsahuje 4 bajty (octety) a může být zapsána ve formátu čtyř desítkových čísel oddělených tečkami (např. 192.168.0.1).
- Každý oktet má hodnotu od 0 do 255.
- IPv4 adresy poskytují přibližně **4,3 miliardy** jedinečných adres, což není dostatečné pro moderní potřeby internetu.

Třídy IPv4 adres

IPv4 adresy jsou rozděleny do tříd na základě prvních bitů:

- **Třída A:** 1.0.0.0 až 126.255.255.255, pro velmi velké sítě. První oktet je síťová část, ostatní tři jsou pro hosty.
- **Třída B:** 128.0.0.0 až 191.255.255.255, pro středně velké sítě. První dva oktety tvoří síťovou část, ostatní dva jsou pro hosty.
- **Třída C:** 192.0.0.0 až 223.255.255.255, pro menší sítě. První tři oktety tvoří síťovou část, poslední oktet je pro hosty.
- **Třída D:** 224.0.0.0 až 239.255.255.255, používá se pro multicast.
- **Třída E:** 240.0.0.0 až 255.255.255.255, rezervována pro experimentální účely.

Typy IPv4 adres

- **Veřejné IP adresy:** Používají se pro připojení k internetu, musí být jedinečné.
- **Privátní IP adresy:** Používají se v lokálních sítích, nelze je směrovat na internet (např. 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8).
- **Speciální adresy:** Např. **127.0.0.1** (loopback adresa) slouží pro testování síťových funkcí na zařízení samotném.

5.3. IPv6 adresa

IPv6 byl vyvinut jako řešení nedostatku adres v IPv4. IPv6 poskytuje větší adresní prostor a lepší vlastnosti pro škálovatelnost a bezpečnost.

Struktura IPv6 adresy

- IPv6 je **128bitová** adresa, což poskytuje přibližně **340 undecilionů (3,4 x 10^38)** jedinečných adres.
- Adresa je zapisována jako osm skupin šestnáctkových čísel oddělených dvojtečkami (např. 2001:0db8:85a3:0000:0000:8a2e:0370:7334).
- IPv6 podporuje zkrácený zápis, kde jsou po sobě jdoucí bloky nul zapsány jako „::“ (např. 2001:0db8:85a3::8a2e:0370:7334).

Typy IPv6 adres

- **Unicast:** Adresuje jedno konkrétní zařízení v síti.
- **Multicast:** Adresuje skupinu zařízení, data posílaná na multicast adresu jsou doručena všem členům této skupiny.
- **Anycast:** Adresa je přiřazena více zařízením, data jsou poslána nejbližšímu zařízení podle směrovacích tabulek.

Základní vlastnosti IPv6

- **Velký adresní prostor:** Zajišťuje, že každé zařízení může mít svou jedinečnou IP adresu.
- **Auto-konfigurace:** IPv6 podporuje automatickou konfiguraci adresy bez potřeby DHCP (SLAAC – Stateless Address Autoconfiguration).
- **Bezpečnost:** IPv6 nativně podporuje IPSec, což zajišťuje lepší zabezpečení komunikace.
- **Efektivnější směrování:** IPv6 zlepšuje směrování díky agregaci adres (CIDR).

5.4. Analýza IP adresy

Analýza IP adresy zahrnuje určení, které části IP adresy odpovídají síti a které části hostovi.
Masky podsítě v IPv4

- Maska podsítě určuje, kolik bitů z IP adresy patří síti a kolik bitů hostům.
- Maska je často zapsána ve formátu CIDR (např. 192.168.1.0/24, kde /24 znamená, že prvních 24 bitů je pro síť).
- Příklad:
    - IP adresa: 192.168.1.10
    - Maska podsítě: 255.255.255.0 (nebo /24)
    - Síťová část: 192.168.1
    - Hostová část: 10
    - Tato adresa je v síti 192.168.1.0 s maskou /24.

Analýza IPv6 adresy

- IPv6 používá podobný koncept jako IPv4, ale díky většímu počtu bitů je adresa rozdělena na 64bitovou síťovou část a 64bitovou část pro identifikaci zařízení (rozhraní).
- Maska je také zapsána ve formátu CIDR, například /64.
- Příklad:
    - IPv6 adresa: 2001:0db8:85a3::8a2e:0370:7334/64
    - Síťová část: 2001:0db8:85a3
    - Identifikátor zařízení: ::8a2e:0370:7334

5.5. DHCP (Dynamic Host Configuration Protocol)

DHCP je protokol, který umožňuje dynamické přidělování IP adres a dalších síťových parametrů zařízením v síti. Tento proces zjednodušuje správu IP adres a minimalizuje možnost konfliktů.
Jak DHCP funguje

1. **DHCPDISCOVER:** Klient, který potřebuje IP adresu, pošle broadcastovou zprávu do sítě, aby našel DHCP server.
2. **DHCPOFFER:** DHCP server odpoví nabídkou IP adresy spolu s dalšími parametry (maska podsítě, brána, DNS server).
3. **DHCPREQUEST:** Klient přijme nabídku a požádá o přidělenou IP adresu.
4. **DHCPACK:** DHCP server potvrdí přiřazení adresy, klient může adresu používat.

Výhody DHCP

- **Automatická konfigurace:** Automaticky přiděluje IP adresy novým zařízením.
- **Dynamické přidělování:** Umožňuje recyklaci adres u zařízení, která nejsou trvale připojena.
- **Centralizovaná správa:** Usnadňuje správu IP adres a síťových nastavení, obzvláště v rozsáhlých sítích.

Parametry, které může DHCP přidělit

- **IP adresa:** Dynamická nebo statická adresa, kterou zařízení dostane.
- **Maska podsítě:** Určuje část IP adresy vyhrazenou pro síť a pro hosty.
- **Výchozí brána (Gateway):** IP adresa směrovače, kterým zařízení komunikuje mimo lokální síť.
- **DNS server:** IP adresa serveru, který zajišťuje převod doménových jmen na IP adresy.
- **Další parametry:** Čas synchronizace (NTP server), doménový název, WINS server apod.

Typy přiřazení DHCP

- **Dynamické přidělování:** IP adresy jsou přidělovány dynamicky z předem definovaného rozsahu a mohou se měnit.
- **Statické přidělování:** Specifické IP adresy jsou přiřazeny konkrétním zařízením na základě jejich MAC adresy.
- **Automatické přidělování:** DHCP server přiřazuje IP adresu zařízení trvale, jakmile ji jednou získá.

5.6. Srovnání IPv4 a IPv6

Funkce	IPv4	IPv6
Délka adresy	32 bitů (4 oktety)	128 bitů (8 bloků)
Zápis	Čtyři desítková čísla oddělená tečkami	Osm šestnáctkových bloků oddělených dvojtečkami
Počet adres	Přibližně 4,3 miliardy	Přibližně 3,4 x 10^38

Funkce	IPv4	IPv6
Podpora auto-konfigurace	Omezená, obvykle s DHCP	SLAAC, podporuje automatickou konfiguraci
Bezpečnost	IPSec je volitelný	IPSec je nativně podporován
Adresy	Unicast, Broadcast, Multicast	Unicast, Anycast, Multicast

5.7. Závěr

IP adresy jsou základem pro identifikaci zařízení v síti. IPv4 je starší a stále široce používaná verze, ale je postupně nahrazována IPv6 kvůli nedostatku adres. Analýza IP adres a rozdělení na síťovou a hostovou část umožňuje efektivní správu sítí. Pro dynamické přidělování IP adres se používá protokol DHCP, který zjednodušuje správu IP adres v síti a minimalizuje chyby způsobené konflikty adres.

6. Protokoly transportní vrstvy

TCP, UDP, ICMP

6.1. Úvod do transportní vrstvy

Transportní vrstva je 4. vrstvou v referenčním modelu OSI a zajišťuje spolehlivý přenos dat mezi dvěma koncovými zařízeními. Je zodpovědná za řízení toku dat, kontrolu chyb a segmentaci dat. Mezi nejpoužívanější protokoly transportní vrstvy patří **TCP (Transmission Control Protocol)** a **UDP (User Datagram Protocol)**. Ačkoliv **ICMP (Internet Control Message Protocol)** patří technicky k síťové vrstvě (3. vrstva OSI), je často zmiňován v souvislosti s transportními protokoly, protože hraje důležitou roli v přenosu řídicích zpráv mezi zařízeními v síti.

6.2. Protokol TCP (Transmission Control Protocol)

TCP je spolehlivý, spojově orientovaný protokol, který zajišťuje doručení dat v pořádku a bez chyb. Je široce využíván v aplikacích, kde je důležité, aby všechna data byla doručena ve správném pořadí, například u webových stránek, e-mailů nebo přenosu souborů.

Vlastnosti TCP

- **Spojově orientovaný protokol:** Před zahájením přenosu dat navazuje spojení pomocí tzv. třícestného navázání spojení (three-way handshake).
- **Spolehlivý přenos:** TCP zajišťuje, že data jsou doručena bez ztráty a v správném pořadí. Pokud je paket ztracen, TCP ho automaticky znovu odešle.
- **Řízení toku (Flow Control):** TCP používá mechanismy pro řízení toku, aby se předešlo přetížení sítě. Děje se tak pomocí okna (window size), které udává, kolik dat může být posláno před tím, než bude vyžadováno potvrzení.
- **Řízení kongesce (Congestion Control):** TCP detekuje přetížení sítě a upravuje rychlost přenosu, aby se předešlo dalším ztrátám paketů.
- **Segmentace a pořadí:** TCP rozděluje data na segmenty a čísluje je. Příjemce pak tyto segmenty skládá do správného pořadí.

Třícestné navázání spojení (Three-Way Handshake)

Před zahájením komunikace TCP navazuje spojení mezi klientem a serverem pomocí tří kroků:

1. **SYN:** Klient pošle požadavek na navázání spojení (SYN) serveru.
2. **SYN-ACK:** Server odpoví potvrzením (SYN-ACK), že je připraven k navázání spojení.
3. **ACK:** Klient potvrdí přijetí odpovědi (ACK), a spojení je navázáno.

Příklady použití TCP

- **HTTP/HTTPS:** Pro přenos webových stránek.
- **FTP:** Pro přenos souborů.
- **SMTP:** Pro odesílání e-mailů.
- **POP3/IMAP:** Pro stahování e-mailů.

Hlavní pole TCP hlavičky

- **Source Port:** Port odesílajícího zařízení.
- **Destination Port:** Port přijímajícího zařízení.
- **Sequence Number:** Určuje pořadí segmentu v rámci toku dat.
- **Acknowledgment Number:** Potvrzuje přijetí dat od druhé strany.
- **Window Size:** Velikost okna pro řízení toku.
- **Checksum:** Kontrola chyb v datech.
- **Flags:** Indikují stav spojení (SYN, ACK, FIN, RST).

6.3. Protokol UDP (User Datagram Protocol)

UDP je nespojový, nespolehlivý protokol, který neposkytuje záruku doručení ani správného pořadí dat. Je vhodný pro aplikace, kde je důležitá rychlost a nízká latence, ale není nezbytné, aby všechna data byla doručena nebo byla doručena ve správném pořadí.

Vlastnosti UDP

- **Nespojový protokol:** UDP nevyžaduje navazování spojení mezi odesílatelem a příjemcem, což snižuje režii a zvyšuje rychlost.

- **Nespolehlivý přenos:** UDP neposkytuje záruku doručení dat ani jejich správného pořadí. Ztracené nebo poškozené pakety nejsou znovu odesílány.
- **Nižší režie:** Díky absenci mechanismů pro kontrolu chyb a zajištění doručení má UDP nižší režii než TCP.
- **Rychlost:** UDP je vhodné pro aplikace, kde je prioritou rychlost přenosu, jako jsou streamování videa nebo VoIP, kde je menší zpoždění důležitější než spolehlivost.

Příklady použití UDP

- **DNS (Domain Name System):** Dotazy na překlad doménových jmen na IP adresy.
- **VoIP (Voice over IP):** Přenos hlasu v reálném čase.
- **Online hry:** Rychlé přenosy dat, kde je lepší ztráta malého množství paketů než zpoždění.
- **Streaming videa:** Zajišťuje nepřerušovaný přenos multimediálního obsahu.

Hlavní pole UDP hlavičky

- **Source Port:** Port odesílajícího zařízení.
- **Destination Port:** Port přijímajícího zařízení.
- **Length:** Celková délka datagramu UDP (hlavička + data).
- **Checksum:** Kontrola chyb v datech (volitelná).

6.4. Protokol ICMP (Internet Control Message Protocol)

ICMP je protokol sítě, který se používá pro zasílání diagnostických a chybových zpráv mezi síťovými zařízeními. ICMP není transportní protokol jako TCP nebo UDP, ale bývá často probírán v této souvislosti, protože umožňuje výměnu řídicích informací o síťové komunikaci.
Vlastnosti ICMP

- **Informativní a diagnostické zprávy:** ICMP umožňuje zařízení v síti vyměňovat informace o stavu sítě, což usnadňuje diagnostiku problémů.
- **Chybová hlášení:** ICMP informuje o problémech při doručování paketů, například pokud paket nemůže být doručen.
- **Nepoužívá porty:** ICMP nemá porty jako TCP nebo UDP, protože neposkytuje přenos dat mezi aplikacemi.

Typy zpráv ICMP

- **Echo Request a Echo Reply:** Používají se při testování dostupnosti zařízení v síti (např. příkaz ping).
- **Destination Unreachable:** Odesláno, pokud paket nemůže být doručen na cílovou adresu (např. pokud není cesta k cíli).
- **Time Exceeded:** Signalizuje, že paket překročil svou maximální dobu životnosti (TTL) při přenosu mezi zařízeními.
- **Redirect:** Informuje zařízení, že existuje lepší cesta k cílové adrese.

Příklady použití ICMP

- **Ping:** Nejznámější aplikace ICMP, která odesílá echo request zprávy k testování dostupnosti cílového zařízení a měření latence.
- **Traceroute:** Používá ICMP (nebo UDP) k zjištění cesty, kterou pakety procházejí mezi zdrojovým a cílovým zařízením. Pomáhá odhalit problémy v síťové trase.

Hlavní pole ICMP hlavičky

- **Type:** Určuje typ ICMP zprávy (např. Echo Request má Type 8, Echo Reply má Type 0).
- **Code:** Kódy rozlišují další podrobnosti o typu zprávy.
- **Checksum:** Slouží ke kontrole chyb v ICMP zprávě.
- **Rest of Header:** Další informace v závislosti na typu zprávy.

6.5. Srovnání TCP, UDP a ICMP

Vlastnost	TCP	UDP	ICMP
Typ protokolu	Spojově orientovaný	Nespojový	Řídicí protokol (spadá do síťové vrstvy)
Spolehlivost	Spolehlivý (zajišťuje doručení a pořadí)	Nespolehlivý (nezajišťuje doručení)	Slouží pouze pro zasílání chyb a informací

Vlastnost	TCP	UDP	ICMP
Kontrola toku	Ano	Ne	Ne
Kontrola chyb	Ano	Ne (kontrola chyb je volitelná)	Ano
Použití	Webové stránky, e-maily, přenos souborů	DNS, VoIP, streaming, online hry	Diagnostika a chybová hlášení
Příklady použití	HTTP, FTP, SMTP	DNS, video streaming, hry	Ping, Traceroute

6.6. Závěr

TCP, UDP a ICMP jsou základní protokoly transportní a síťové vrstvy, které zajišťují různé aspekty komunikace v počítačových sítích. TCP poskytuje spolehlivý, spojově orientovaný přenos dat s kontrolou chyb a řízením toku, zatímco UDP je vhodný pro aplikace, kde je prioritou rychlost a nízká režie. ICMP slouží k diagnostice a odesílání chybových zpráv a je zásadní pro monitorování stavu sítě. Výběr vhodného protokolu závisí na požadavcích aplikace na spolehlivost a rychlost komunikace.

7. Návrh lokální sítě

Metody propojení, parametry, postup návrhu, konfigurace OS (MS Windows a Linux)

7.1. Úvod do návrhu lokální sítě (LAN)

Lokální síť (LAN – Local Area Network) je síť, která propojuje zařízení na omezeném geografickém území, typicky v jedné budově nebo komplexu. Návrh LAN je klíčový pro efektivní a bezpečný přenos dat mezi zařízeními, zajištění dostatečné šířky pásma a minimalizaci výpadků.
Správný návrh LAN zahrnuje výběr vhodných propojení, konfiguraci síťových prvků a nastavení zařízení, aby vyhovovala požadavkům uživatelů a aplikací.

7.2. Metody propojení v LAN

Existuje několik metod propojení zařízení v LAN, mezi které patří:
a) Kabelové propojení

- **Typy kabelů:** Nejčastěji se používají kroucené dvojlinky (UTP/STP) nebo optická vlákna.
- **Topologie:** Kabelové propojení typicky vytváří hvězdicovou topologii (každé zařízení je připojeno k centrálnímu přepínači).
- **Rychlost:** Standardně 1 Gb/s až 10 Gb/s, v případě optických vláken i více.
- **Výhody:** Stabilita, spolehlivost, vysoká přenosová rychlost, odolnost vůči rušení.
- **Nevýhody:** Omezená mobilita, náročná instalace.

b) Bezdrátové propojení (Wi-Fi)

- **Frekvence:** 2,4 GHz, 5 GHz a nově 6 GHz (pro Wi-Fi 6E).
- **Topologie:** Většinou hvězdicová, s centrálním přístupovým bodem (Access Point).
- **Rychlost:** Standardně 300 Mb/s až 1 Gb/s, záleží na standardu (802.11n, 802.11ac, 802.11ax).
- **Výhody:** Mobilita, snadná instalace, žádné fyzické kabely.
- **Nevýhody:** Náchylnost k rušení, nižší stabilita a rychlost než u kabelového propojení.

c) Hybridní propojení

- Kombinace kabelového a bezdrátového propojení, často využívané ve větších sítích, kde se kabelové připojení používá pro páteřní síť (backbone) a bezdrátové propojení pro koncová zařízení.

7.3. Parametry lokální sítě

Při návrhu LAN je nutné zvážit následující parametry:
a) Šířka pásma

- Dostatečná šířka pásma je nezbytná pro rychlý přenos dat mezi zařízeními, zejména pro aplikace náročné na přenosy (streamování, VoIP, přenos souborů).

b) Topologie sítě

- **Hvězdicová:** Každé zařízení je připojeno k centrálnímu přepínači, což zjednodušuje správu a údržbu.
- **Sběrnicová:** Jedno kabelové spojení propojuje všechna zařízení, ale tato topologie je méně běžná.
- **Stromová nebo hybridní:** Kombinace hvězdicové a sběrnicové topologie, vhodná pro rozsáhlé sítě.

c) Redundance a dostupnost

- Pro zajištění nepřerušeného provozu by měla síť obsahovat záložní cesty a prvky, které minimalizují dopad selhání komponent.

d) Bezpečnost

- Ochrana proti neoprávněnému přístupu, šifrování bezdrátové sítě (např. WPA3), firewall, VLAN pro oddělení sítě.

e) Počet a typ zařízení

- Počet zařízení ovlivňuje kapacitu a výkon sítě. Je třeba vzít v úvahu počet uživatelů, jejich zařízení a typ síťových aplikací.

7.4. Postup návrhu LAN

Krok 1: Analýza požadavků

- Určete požadavky na přenos dat, počet uživatelů, typy aplikací, požadavky na šířku pásma, bezpečnost a zálohování.

Krok 2: Návrh topologie

- Zvolte vhodnou topologii s ohledem na počet zařízení, vzdálenosti a možnosti škálování.

Krok 3: Výběr síťových prvků

- Zvolte přepínače, routery, přístupové body a kabeláž podle parametrů sítě. Rozhodněte se, zda použít managed nebo unmanaged switche, a zda je potřeba Power over Ethernet (PoE) pro napájení zařízení.

Krok 4: Rozdělení sítě (VLAN)

- Pro zvýšení bezpečnosti a efektivity provozu rozdělte síť na VLANy, což umožňuje logické oddělení provozu (např. administrativní VLAN, síť pro návštěvníky, síť pro IoT zařízení).

Krok 5: Konfigurace IP adresace

- Rozvrhněte adresní prostor, přidělte statické IP adresy kritickým zařízením (servery, síťové tiskárny) a nastavte DHCP pro dynamické přidělování IP adres koncovým zařízením.

Krok 6: Zabezpečení

- Implementujte firewall, nastavte přístupová práva, povolte šifrování na bezdrátových přístupových bodech, rozdělte síť na VLANy, a pokud je potřeba, nastavte přístupové body pomocí WPA3 nebo 802.1X.

Krok 7: Testování a optimalizace

- Otestujte síť na přenosové rychlosti, odezvu a stabilitu. Upravte konfiguraci, pokud zjistíte problémy s přetížením nebo výpadky.

7.5. Konfigurace OS v LAN (MS Windows a Linux)

Konfigurace sítě v MS Windows

1. Nastavení IP adresy (statické nebo dynamické)
    - Otevřete Nastavení > Síť a internet > Ethernet (nebo Wi-Fi).
    - Klikněte na **Změnit možnosti adaptéru** > pravým tlačítkem na adaptér > **Vlastnosti**.
    - Zvolte Protokol IP verze 4 (TCP/IPv4) > Vlastnosti.
    - Nastavte **získání adresy IP** automaticky (pro DHCP) nebo zadejte statickou IP adresu, masku podsítě, výchozí bránu a DNS servery.
2. Připojení k síti
    - U Wi-Fi vyberte dostupnou síť a zadejte heslo.
    - Pro Ethernetové připojení by mělo zařízení automaticky získat IP adresu z DHCP serveru, pokud je povoleno.
3. Nastavení pracovních skupin nebo domény
    - Pro připojení k pracovní skupině nebo doméně: Ovládací panely > Systém a zabezpečení > Systém > Změnit nastavení.
    - Zadejte název pracovní skupiny nebo domény.
4. Konfigurace firewallu
    - Otevřete **Windows Defender Firewall** v Ovládacích panelech.
    - Upravte pravidla pro příchozí a odchozí připojení podle potřeby.
5. Příkazový řádek
    - **ipconfig:** Zobrazí IP adresu a konfiguraci síťových adaptérů.
    - **ping [IP adresa]:** Testuje dostupnost jiného zařízení v síti.
    - **netstat:** Zobrazuje aktivní spojení a porty.
    - **tracert [IP adresa]:** Ukazuje trasu, kterou data putují k cílovému zařízení.

Konfigurace sítě v Linuxu

1. Nastavení IP adresy (statické nebo dynamické)
    - **Network Manager (grafické prostředí):** Otevřete Nastavení sítě, vyberte síťový adaptér a nastavte IP adresu ručně nebo pomocí DHCP.
    - Příkazový řádek (CLI):
        - Dynamická IP: sudo dhclient [interface]
        - Statická IP:

```
sudo ip addr add 192.168.1.10/24 dev eth0
sudo ip route add default via 192.168.1.1
```

2. Konfigurace DNS
    - Pro ruční nastavení DNS serverů v Linuxu můžete upravit soubor `/etc/resolv.conf`:

```
sudo nano /etc/resolv.conf
nameserver 8.8.8.8
nameserver 8.8.4.4
```

3. Nastavení síťových služeb (DHCP klient nebo server)
    - **DHCP klient:** Mnoho distribucí automaticky používá DHCP klienta. Pro přidělení adresy DHCP z příkazového řádku:

```
sudo dhclient eth0
```

    - **DHCP server:** V případě, že chcete na Linuxu nastavit DHCP server, musíte nainstalovat příslušný software (např. isc-dhcp-server) a upravit konfiguraci `/etc/dhcp/dhcpd.conf`.
4. Firewall – iptables nebo firewalld
    - **iptables:** Slouží ke správě pravidel firewallu:

```
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT  # Povolí SSH
```

    - firewalld (na distribucích jako CentOS nebo Fedora):

```
sudo firewall-cmd --zone=public --add-port=80/tcp --permanent
sudo firewall-cmd --reload
```

7.6. Síťové nástroje pro správu a diagnostiku

- **`ifconfig` nebo `ip addr`:** Zobrazí informace o síťových adaptérech.
- **`ping` [IP adresa]:** Testuje dostupnost zařízení v síti.
- **`traceroute` [IP adresa]:** Ukazuje trasu k cílovému zařízení.
- **`netstat` nebo `ss`:** Zobrazuje aktivní spojení a porty.
- **`nmap`:** Skenuje síťové porty a zařízení (vyžaduje instalaci).

7.7. Závěr

Návrh lokální sítě vyžaduje detailní analýzu požadavků, správné plánování topologie, výběr vhodných zařízení a konfiguraci síťových parametrů. Metody propojení mohou být kabelové, bezdrátové nebo hybridní. Po návrhu fyzické sítě je důležité správně nastavit síťové parametry a zabezpečení na jednotlivých operačních systémech, což zahrnuje konfiguraci IP adres, připojení k síti, nastavení firewallu a další síťové nástroje.

8. Routování

Princip routování, používané protokoly, NAT

8.1. Úvod do routování

Routování je proces, kterým síťové zařízení, obvykle **router** (směrovač), přenáší data mezi různými sítěmi. Smyslem routování je určit nejefektivnější cestu, kterou by měl datový paket urazit, aby dosáhl svého cílového zařízení. Routery analyzují cílovou IP adresu paketu a na základě směrovacích tabulek rozhodují, kam paket předat.
Routování je klíčové pro propojení různých sítí, zejména pokud jde o komunikaci mezi lokálními sítěmi (LAN) a širšími sítěmi, jako je internet (WAN).

8.2. Princip routování

Routování funguje na základě směrovacích tabulek, které obsahují informace o různých sítích a cestách k nim. Router zjišťuje nejlepší cestu pro každý paket pomocí těchto tabulek a směrovacích protokolů.
Proces routování

1. **Analýza cílové adresy:** Router přečte cílovou IP adresu v hlavičce paketu.
2. **Hledání směrovací cesty:** Router porovná cílovou adresu s položkami ve směrovací tabulce, aby zjistil nejvhodnější cestu.
3. **Forwardování paketu:** Na základě záznamu ve směrovací tabulce router pošle paket na další síťové zařízení nebo přímo na cílové zařízení.
4. **Aktualizace směrovacích tabulek:** Směrovací tabulky se mohou aktualizovat dynamicky pomocí směrovacích protokolů nebo být nastaveny staticky správcem sítě.

Typy routování

- **Statické routování:** Směrovací cesty jsou nastavovány manuálně správcem. Je vhodné pro malé a jednoduché sítě, kde není potřeba měnit směrování často.
- **Dynamické routování:** Směrovací tabulky se automaticky aktualizují pomocí směrovacích protokolů, což umožňuje adaptaci na změny v síti. Vhodné pro složité a velké sítě.

8.3. Používané směrovací protokoly

Směrovací protokoly se dělí na dvě základní kategorie: **vnitřní směrovací protokoly (IGP – Interior Gateway Protocols)**, které se používají uvnitř autonomního systému (AS), a **vnější směrovací protokoly (EGP – Exterior Gateway Protocols)**, které propojují různé autonomní systémy.
A) Vnitřní směrovací protokoly (IGP)

1. RIP (Routing Information Protocol)
    - **Verze:** RIP v1 a RIP v2
    - **Metoda:** Protokol vektor vzdáleností, používá počet skoků (hops) jako metriku.
    - **Vlastnosti:** Jednoduchý, používá max. 15 skoků, což omezuje jeho použití na malé sítě.
    - **Nevýhody:** Pomalá konvergence (delší čas pro aktualizaci tabulek), omezený počet skoků, což omezuje jeho použití ve větších sítích.
2. OSPF (Open Shortest Path First)
    - **Typ:** Protokol stavu linky (link-state).
    - **Metoda:** Používá Dijkstrův algoritmus pro výpočet nejkratší cesty.
    - **Vlastnosti:** Podporuje rychlou konvergenci, vhodný pro větší a složité sítě, rozděluje síť na oblasti (area) pro lepší škálovatelnost.
    - **Výhody:** Rychlá reakce na změny v síti, není omezen počtem skoků jako RIP.
3. EIGRP (Enhanced Interior Gateway Routing Protocol)
    - **Typ:** Hybridní protokol (kombinace vektor vzdáleností a stavu linky).
    - **Vlastnosti:** Vyvinutý společností Cisco, poskytuje rychlejší konvergenci než RIP, zajišťuje redundanci a podporuje vyvážení zátěže (load balancing).
    - **Výhody:** Schopnost podporovat více síťových protokolů (IPv4 a IPv6), vhodný pro středně velké a velké sítě.
4. IS-IS (Intermediate System to Intermediate System)
    - **Typ:** Protokol stavu linky (link-state).
    - **Metoda:** Používá algoritmus Shortest Path First (SPF) pro výpočet nejkratší cesty.

- **Vlastnosti:** Vyvinutý pro použití v rámci OSI modelu, ale podporuje i IPv4 a IPv6. IS-IS pracuje na úrovni 2. vrstvy OSI modelu, což umožňuje jeho rychlé směrování bez závislosti na IP adresách.
- **Výhody:** Vhodný pro velmi velké sítě, podporuje hierarchické rozdělení sítě na oblasti (podobně jako OSPF), široce používán v páteřních sítích poskytovatelů internetu.
- **Nevýhody:** Méně rozšířený než OSPF, ale oblíbený v prostředí poskytovatelů internetových služeb kvůli flexibilitě a efektivitě.

B) Vnější směrovací protokoly (EGP)

1. BGP (Border Gateway Protocol)
   - **Typ:** Protokol vektor vzdáleností s podporou cesty (path-vector).
   - **Vlastnosti:** Používá se k propojení různých autonomních systémů na internetu, směruje pakety mezi ISP a velkými podniky.
   - **Výhody:** Schopnost pracovat s velkými sítěmi, rozšiřitelný, důležitý pro globální routování na internetu.
   - **Nevýhody:** Složitá konfigurace a správa, vyžaduje vyšší úroveň technických znalostí.

8.4. NAT (Network Address Translation)

NAT je technologie používaná k překladu IP adres mezi soukromou (lokální) a veřejnou sítí. NAT umožňuje, aby více zařízení v lokální síti používalo jednu veřejnou IP adresu k přístupu na internet, což šetří IP adresy a zvyšuje bezpečnost.
Typy NAT

1. Statický NAT
   - Překládá konkrétní privátní IP adresu na konkrétní veřejnou IP adresu.
   - Využití: Používá se, pokud je potřeba mít trvalý přístup z internetu k zařízení v privátní síti (např. servery).
2. Dynamický NAT
   - Překládá privátní IP adresy na dynamicky přidělené veřejné IP adresy z předem definovaného rozsahu.
   - Využití: Používá se, pokud je potřeba, aby privátní zařízení mělo přístup k internetu, ale není potřeba pevná IP adresa.
3. PAT (Port Address Translation) – Overloading NAT
   - Překládá více privátních IP adres na jednu veřejnou IP adresu pomocí rozdílných portů.
   - Využití: Nejčastější typ NAT, běžně používaný u domácích routerů, umožňuje více zařízením sdílet jednu veřejnou IP adresu pomocí různých portů.

Výhody NAT

- **Úspora veřejných IP adres:** Snižuje počet veřejných IP adres potřebných pro připojení zařízení k internetu.
- **Zvýšená bezpečnost:** Skryje privátní IP adresy zařízení uvnitř sítě před internetem, což znesnadňuje jejich přímý přístup zvenčí.
- **Jednoduchá konfigurace:** NAT je často snadno konfigurovatelný na routerech a firewallech.

Nevýhody NAT

- **Kompatibilita s některými protokoly:** NAT může způsobit problémy u některých protokolů, které embedují IP adresy do paketů (např. FTP).
- **Problémy s end-to-end komunikací:** NAT může zkomplikovat přímou komunikaci mezi dvěma zařízeními, například při použití VoIP nebo P2P aplikací.

8.5. Příklady nastavení NAT

Nastavení NAT na domácím routeru (PAT)
Typickým nastavením NAT na domácím routeru je **PAT (Overloading NAT)**, kde router překládá více zařízení na jednu veřejnou IP adresu s různými porty:

1. Router přiděluje každému zařízení v lokální síti privátní IP adresu.
2. Při pokusu o přístup na internet router přeloží privátní IP adresu na veřejnou adresu, ale udržuje si tabulku překladu, kde sleduje spojení na základě portů.
3. Odpovědi z internetu jsou pak přesměrovány zpět na příslušná zařízení pomocí uložených portových čísel.

Nastavení statického NAT
Statický NAT se používá pro mapování jednoho zařízení v lokální síti na konkrétní veřejnou IP adresu. Například, pokud chceme, aby server v lokální síti byl přístupný z internetu:

1. Nastavíme statické mapování IP adresy interního serveru na veřejnou IP adresu.

2. Veškerý provoz na veřejnou IP adresu je přesměrován na interní server.

8.6. Závěr

Routování je zásadním prvkem pro propojení a komunikaci mezi sítěmi. Směrovací protokoly, jako jsou RIP, OSPF, IS-IS a BGP, umožňují routerům dynamicky aktualizovat směrovací tabulky a optimalizovat trasy pro přenos dat. NAT je technologie, která šetří veřejné IP adresy a zvyšuje bezpečnost sítě tím, že skryje privátní adresy zařízení. Pochopení principů routování, směrovacích protokolů a správné implementace NAT je klíčové pro správu sítí.

9. DNS

Význam, položky DNS záznamu, popis protokolu

9.1. Význam DNS (Domain Name System)

DNS (Domain Name System) je systém, který překládá doménová jména (např. www.example.com) na IP adresy (např. 192.0.2.1) a naopak. Tento proces umožňuje uživatelům přistupovat k webovým stránkám a službám na internetu pomocí snadno zapamatovatelných jmen, zatímco síťová zařízení pracují s IP adresami. DNS se často označuje jako „telefonní seznam internetu“, protože překládá čitelná doménová jména na adresy potřebné pro komunikaci v síti.
DNS je kritickou součástí internetu, protože zajišťuje dostupnost služeb a umožňuje rychlé a spolehlivé spojení mezi klientem a serverem. DNS používá hierarchickou strukturu, která je rozčleněna do různých doménových úrovní (např. kořenová doména, TLD, druhá úroveň).

9.2. Položky DNS záznamu

DNS systém pracuje se záznamy, které obsahují konkrétní informace o doméně a jsou uloženy na DNS serverech. Mezi základní typy DNS záznamů patří:
a) A (Address) záznam

- **Účel:** Překládá doménové jméno na IPv4 adresu.
- **Příklad:** „example.com“ → 192.0.2.1
- **Použití:** Zajišťuje spojení na konkrétní server při zadání doménového jména.

b) AAAA (IPv6 Address) záznam

- **Účel:** Překládá doménové jméno na IPv6 adresu.
- **Příklad:** „example.com“ → 2001:0db8:85a3:0000:0000:8a2e:0370:7334
- **Použití:** Zajišťuje spojení na konkrétní server při zadání doménového jména s podporou IPv6.

c) CNAME (Canonical Name) záznam

- **Účel:** Přesměrovává doménové jméno na jiné doménové jméno (alias).
- **Příklad:** „www.example.com“ → „example.com“
- **Použití:** Usnadňuje správu více domén, které vedou na stejnou IP adresu nebo server.

d) MX (Mail Exchange) záznam

- **Účel:** Určuje poštovní server, který zpracovává e-maily pro danou doménu.
- **Příklad:** „example.com“ → mail.example.com s prioritou 10
- **Použití:** Umožňuje nastavení různých poštovních serverů a priorit, což zajišťuje doručování e-mailů do správné schránky.

e) PTR (Pointer) záznam

- **Účel:** Překládá IP adresu zpět na doménové jméno (reverse DNS lookup).
- **Příklad:** 192.0.2.1 → „example.com“
- **Použití:** Ověřuje důvěryhodnost a identitu serveru, často používané u e-mailových serverů.

f) TXT (Text) záznam

- **Účel:** Obsahuje libovolné textové informace o doméně, které mohou být použity pro různé účely.
- **Příklad:** SPF nebo DKIM záznamy pro e-mailovou autentizaci, informace o vlastnictví domény.
- **Použití:** Zabezpečení e-mailové komunikace, ověření domény pro různé služby (např. Google, Microsoft).

g) NS (Name Server) záznam

- **Účel:** Určuje autoritativní DNS servery pro danou doménu.
- **Příklad:** „example.com“ → ns1.example.com
- **Použití:** Definuje, které servery jsou zodpovědné za DNS záznamy domény a kde jsou uloženy.

h) SOA (Start of Authority) záznam

- **Účel:** Obsahuje základní informace o doméně, včetně primárního DNS serveru, e-mailu správce domény, sériového čísla a nastavení pro cache.
- **Příklad:** SOA záznam pro „example.com“ určuje primární DNS server ns1.example.com.
- **Použití:** Pomáhá spravovat aktualizace DNS záznamů a určovat, jak dlouho jsou cacheovány.

i) SRV (Service) záznam

- **Účel:** Specifikuje umístění konkrétní služby (např. SIP, XMPP) v rámci domény.
- **Příklad:** Definuje server pro VoIP službu v doméně.
- **Použití:** Umožňuje připojení ke konkrétním službám na správné IP a portu.

9.3. Popis DNS protokolu

DNS protokol umožňuje dotazy na překládání doménových jmen na IP adresy a zajišťuje správnou odpověď z DNS serverů. Pracuje na **aplikační vrstvě** a používá pro komunikaci protokol **UDP** na portu 53, případně **TCP** pro přenos větších dat (např. pro zónový přenos).
Princip fungování DNS

1. **Dotaz klienta:** Když uživatel zadá adresu webové stránky, klient (např. prohlížeč) vyšle DNS dotaz k překladu doménového jména na IP adresu.
2. **Kontrola cache:** DNS klient nejprve zkontroluje, zda již nemá adresu uloženou v cache. Pokud ano, použije ji přímo.
3. **Rekurzivní dotaz:** Pokud není odpověď v cache, klient se obrátí na lokální DNS server (rekurzivní resolver), který se dotazuje dál.
4. **Iterativní dotazy:** Rekurzivní resolver postupně dotazuje autoritativní servery (kořenové servery, TLD servery a autoritativní servery pro doménu) a získává odpověď.
5. **Odpověď klientovi:** Rekurzivní resolver vrátí klientovi IP adresu, kterou následně prohlížeč použije pro připojení k serveru.

Typy DNS dotazů

- **Rekurzivní dotaz:** Požadavek na úplnou odpověď (přesný záznam), kterou musí DNS server najít. Rekurzivní resolver hledá IP adresu ve všech úrovních DNS systému.
- **Iterativní dotaz:** DNS server poskytne klientovi, pokud nemá odpověď, pouze odkaz na další server, který může mít potřebné informace. Klient musí sám postupně projít celou cestu.
- **Inverzní dotaz (Reverse Lookup):** Opačný proces k překladu IP adres na doménová jména (využívá PTR záznamy).

Typy DNS odpovědí

- **Autoritativní odpověď:** Přesná odpověď od DNS serveru, který je primárním zdrojem informací pro doménu.
- **Neautoritativní odpověď:** Odpověď z DNS cache jiného serveru, která může být uložena z předchozích dotazů.
- **Odpověď „nenalezeno“:** Odpověď v případě, že doménové jméno neexistuje.

Bezpečnostní rozšíření – DNSSEC (DNS Security Extensions)
DNSSEC přidává k DNS ověřovací mechanismy, které zajišťují, že odpovědi DNS serverů nebyly změněny nebo podvrženy.

- **Ověřování:** DNSSEC ověřuje autenticitu DNS odpovědí pomocí digitálních podpisů.
- **Klíče:** Používá asymetrické klíče k podpisu a ověření DNS záznamů.
- **Ochrana proti útokům:** Chrání před útoky, jako je DNS cache poisoning, kde útočník vloží falešné DNS záznamy do cache.

Výhody DNS protokolu

- **Efektivní vyhledávání:** DNS umožňuje rychlé a hierarchické vyhledávání IP adres.
- **Hierarchická struktura:** Umožňuje rozdělit zátěž a zvyšuje redundanci, což zajišťuje vysokou dostupnost.
- **Možnost cacheování:** Pomáhá snižovat latenci, protože záznamy mohou být uloženy v cache na DNS serverech i klientech.

Nevýhody a rizika DNS protokolu

- **Bezpečnostní rizika:** DNS může být zranitelný vůči útokům, jako jsou DNS cache poisoning a DDoS útoky.
- **Závislost na externích serverech:** DNS dotazy mohou být zpomaleny, pokud jsou servery mimo provoz nebo přetíženy.

9.4. Závěr

DNS je klíčovým systémem pro fungování internetu, který zajišťuje překlad doménových jmen na IP adresy a umožňuje uživatelům pohodlný přístup ke službám. DNS pracuje s několika druhy záznamů, které definují IP adresy, poštovní servery, aliasy a další informace o doménách. DNS protokol zahrnuje mechanismy pro efektivní a rychlé dotazování pomocí rekurzivních a iterativních dotazů. Pro zvýšení bezpečnosti byl vyvinut DNSSEC, který chrání DNS záznamy před podvržením a zajišťuje autenticitu odpovědí.

10. Elektronická pošta

Princip, protokoly, struktura e-mailu

10.1. Princip elektronické pošty (e-mailu)

Elektronická pošta, zkráceně e-mail, je způsob komunikace, který umožňuje odesílání zpráv mezi uživateli prostřednictvím internetu nebo jiné počítačové sítě. Zpráva e-mailu se skládá z textu a může obsahovat také přílohy, jako jsou dokumenty nebo obrázky. E-mailový systém se skládá z klientů (e-mailových aplikací) a serverů, které zajišťují doručování a správu zpráv.

Princip fungování e-mailu

1. **Vytvoření a odeslání zprávy:** Uživatel vytvoří e-mail a odešle jej přes svůj e-mailový klient.
2. **Komunikace mezi servery:** E-mailový klient komunikuje s odchozím poštovním serverem (SMTP server), který zajišťuje doručení zprávy do cílového serveru.
3. **Doručení zprávy:** Příchozí poštovní server příjemce (IMAP nebo POP3 server) přijme e-mail a uloží jej do schránky příjemce.
4. **Stažení zprávy:** Příjemce zprávu stáhne nebo zobrazí prostřednictvím svého e-mailového klienta.

10.2. E-mailové protokoly

E-mailový systém používá různé protokoly, které zajišťují komunikaci mezi klientem a serverem, a mezi jednotlivými servery. Nejčastěji používané protokoly jsou **SMTP, IMAP** a **POP3**.

A) SMTP (Simple Mail Transfer Protocol)

- **Účel:** SMTP je protokol používaný pro odesílání e-mailů z klienta na server a pro přenos mezi servery.
- **Porty:** Standardně využívá port 25 (nezabezpečené připojení), 465 (SMTP s SSL) nebo 587 (SMTP s TLS).
- **Princip fungování:** SMTP pracuje na bázi „store-and-forward“, což znamená, že e-mail je odeslán z klienta na server, který jej následně předá do cílové schránky nebo na jiný server, pokud je to potřeba.
- **Použití:** SMTP je základním protokolem pro odesílání e-mailů, ale ne pro příjem a čtení zpráv.

B) IMAP (Internet Message Access Protocol)

- **Účel:** IMAP slouží k přístupu k e-mailům uloženým na serveru z různých zařízení. Na rozdíl od POP3 ponechává zprávy na serveru, což umožňuje synchronizaci mezi zařízeními.
- **Porty:** Používá port 143 (nezabezpečené připojení) nebo 993 (IMAP s SSL/TLS).
- **Princip fungování:** Uživatel může číst e-maily, aniž by musel zprávy stahovat, což umožňuje lepší správu a synchronizaci e-mailů mezi více zařízeními.
- **Výhody:** IMAP umožňuje udržovat stav e-mailů na serveru (např. označení jako přečtené) a přístup z různých zařízení.

C) POP3 (Post Office Protocol version 3)

- **Účel:** POP3 umožňuje stahování e-mailů ze serveru na lokální zařízení a obvykle odstraňuje zprávy ze serveru po stažení.
- **Porty:** Používá port 110 (nezabezpečené připojení) nebo 995 (POP3 s SSL/TLS).
- **Princip fungování:** POP3 je jednodušší protokol než IMAP, typicky stahuje zprávy na klienta a odstraňuje je ze serveru, což omezuje možnost synchronizace mezi zařízeními.
- **Nevýhody:** Nepodporuje synchronizaci mezi zařízeními, což znamená, že změny provedené na jednom zařízení se neprojeví na jiném.

D) Protokoly pro zabezpečení (SSL/TLS)

- **SSL (Secure Sockets Layer)** a **TLS (Transport Layer Security)** jsou protokoly, které zajišťují šifrovanou komunikaci mezi klientem a serverem. Protokoly jako IMAP, POP3 a SMTP často používají zabezpečené verze (IMAPS, POP3S, SMTPS) pro ochranu přenášených dat.

10.3. Struktura e-mailu

E-mail má specifickou strukturu, která se skládá ze dvou hlavních částí: **hlavičky** a **těla zprávy**. Může také obsahovat přílohy.

A) Hlavička e-mailu

Hlavička obsahuje metadata o zprávě a informace nezbytné pro doručení e-mailu.

- **From:** Obsahuje adresu odesílatele.

- **To:** Obsahuje adresu příjemce (nebo více příjemců).
- **Cc (Carbon Copy):** Obsahuje adresy osob, které dostanou kopii e-mailu. Tito příjemci vidí ostatní adresáty v poli Cc.
- **Bcc (Blind Carbon Copy):** Obsahuje adresy osob, které dostanou kopii e-mailu, ale jejich adresy nejsou viditelné ostatním příjemcům.
- **Subject:** Předmět e-mailu, krátký popis obsahu zprávy.
- **Date:** Datum a čas odeslání e-mailu.
- **Message-ID:** Unikátní identifikátor zprávy, který je generován odesílajícím serverem. Používá se pro sledování e-mailů a jejich vláken.
- **Reply-To:** Určuje e-mailovou adresu, na kterou má být odpověď zaslána, pokud se liší od adresy v poli From.

B) Tělo e-mailu
Tělo zprávy obsahuje hlavní obsah e-mailu. Tělo může být ve formátu **plain text** (prostý text) nebo **HTML** (umožňuje formátování, obrázky, odkazy apod.).

- **Plain text:** Jednoduchý text bez formátování.
- **HTML:** Umožňuje použití formátovaného textu, stylů, obrázků a odkazů. HTML zprávy jsou čitelnější a atraktivnější, ale mohou být zneužity k šíření škodlivého kódu.

C) Přílohy
Přílohy jsou součástí e-mailu, která umožňuje přenos souborů, jako jsou obrázky, dokumenty nebo jiné soubory.

- **MIME (Multipurpose Internet Mail Extensions):** Standard pro odesílání příloh, který definuje, jak jsou soubory vkládány a kódovány, aby byly přenositelné prostřednictvím e-mailu.
- **Bezpečnostní rizika:** Přílohy mohou obsahovat malware, proto je důležité, aby e-mailové servery skenovaly přílohy a uživatelé byli opatrní při otevírání souborů z neznámých zdrojů.

10.4. Zabezpečení elektronické pošty

E-mailový provoz je často terčem kybernetických útoků, a proto je třeba dbát na bezpečnost.
A) SPF (Sender Policy Framework)

- SPF je autentizační metoda, která zabraňuje padělání e-mailových adres tím, že určuje, které servery mohou odesílat e-maily jménem domény odesílatele. Tím chrání doménu před phishingem a spamem.

B) DKIM (DomainKeys Identified Mail)

- DKIM umožňuje odesílateli e-mailu připojit k zprávě digitální podpis, který příjemce ověří pomocí veřejného klíče v DNS záznamu odesílatele. Zajišťuje integritu zprávy a pomáhá odhalit podvržené e-maily.

C) DMARC (Domain-based Message Authentication, Reporting & Conformance)

- DMARC je standard, který kombinuje SPF a DKIM a umožňuje doménám určit, jak se mají příjemci chovat k podezřelým e-mailům (např. odmítnout, označit jako spam). DMARC také poskytuje reporty o stavu e-mailů odeslaných doménou, což pomáhá chránit před phishingem.

D) Šifrování

- Pro zabezpečení obsahu e-mailu se používá šifrování (např. pomocí S/MIME nebo PGP). Šifrování zajišťuje, že e-mailový obsah mohou číst pouze odesílatel a zamýšlený příjemce, a chrání před neoprávněným přístupem během přenosu.

10.5. Výhody a nevýhody e-mailu

Výhody

- **Rychlost a dostupnost:** Umožňuje téměř okamžité odesílání zpráv po celém světě.
- **Nízké náklady:** Posílání e-mailů je levné, zejména ve srovnání s klasickou poštou.
- **Možnost příloh:** Uživatelé mohou snadno přenášet soubory a multimédia.
- **Historie komunikace:** E-mail umožňuje snadno sledovat historii komunikace a uchovávat záznamy o důležitých zprávách.

Nevýhody

- **Bezpečnostní rizika:** E-mail je náchylný k útokům, jako je phishing, spam a malware v přílohách.
- **Nedostatečné soukromí:** Bez šifrování může být obsah e-mailu přečten během přenosu nebo na serverech.

- **Závislost na internetu:** Pro odesílání a přijímání e-mailů je nutné internetové připojení.

10.6. Závěr

Elektronická pošta je základním nástrojem pro digitální komunikaci, který umožňuje rychlý a efektivní přenos zpráv a souborů. Princip e-mailu spočívá v přenosu zpráv mezi servery a klienty prostřednictvím protokolů SMTP, IMAP a POP3. Struktura e-mailu obsahuje hlavičku, tělo a přílohy, které lze kódovat pomocí MIME. K zabezpečení e-mailů se využívají technologie jako SPF, DKIM, DMARC a šifrování, které zajišťují autenticitu a důvěryhodnost zpráv. E-mail zůstává jedním z nejpoužívanějších komunikačních prostředků navzdory některým bezpečnostním rizikům.

11. FTP a sdílení souborů

Princip, protokoly, zabezpečení

11.1. Princip FTP a sdílení souborů

FTP (File Transfer Protocol) je jeden z nejstarších a nejrozšířenějších protokolů pro přenos a sdílení souborů přes síť. Tento protokol umožňuje uživatelům nahrávat, stahovat a spravovat soubory uložené na vzdálených serverech.
Princip FTP
FTP využívá klient-server architekturu:

1. **Klient** se připojí k **FTP serveru**, který poskytuje přístup k souborům.
2. Uživatel může na serveru vykonávat různé operace, jako je nahrávání, stahování a mazání souborů, vytváření a správa složek.
3. Pro připojení je nutné zadat přihlašovací údaje (uživatelské jméno a heslo) nebo lze použít anonymní přístup, pokud server umožňuje anonymní FTP.

FTP komunikuje přes dvě samostatná spojení mezi klientem a serverem:

- **Řídicí spojení** (Control Connection): Používá se pro příkazy a odpovědi mezi klientem a serverem. Využívá port 21.
- **Datové spojení** (Data Connection): Používá se pro přenos souborů. V závislosti na typu připojení může být otevřeno na různých portech.

11.2. Protokoly pro sdílení souborů

Kromě FTP existují další protokoly pro sdílení souborů. Zde jsou nejčastější protokoly a jejich vlastnosti:
A) FTP (File Transfer Protocol)

- **Porty:** Řídicí spojení na portu 21, datové spojení na různých portech (20 pro aktivní režim).
- Režimy přenosu:
  - **Aktivní režim:** Klient otevírá port pro datové spojení a server se k němu připojuje.
  - **Pasivní režim:** Server otevře port pro datové spojení a klient se k němu připojí (vhodné pro sítě s firewally).
- **Nevýhody:** FTP není šifrovaný, což znamená, že přihlašovací údaje a data jsou přenášeny v otevřeném textu a mohou být zachyceny.

B) FTPS (FTP Secure)

- **Účel:** Bezpečnější verze FTP, která přidává šifrování pomocí protokolů SSL/TLS.
- **Porty:** Používá port 990 pro šifrované spojení a běžně porty 989 a 990 pro data.
- **Výhody:** Zajišťuje šifrování dat a přihlašovacích údajů, což chrání proti odposlechu a útokům typu „man-in-the-middle“.

C) SFTP (SSH File Transfer Protocol)

- **Účel:** Protokol pro bezpečný přenos souborů přes SSH (Secure Shell).
- **Port:** Používá port 22 (stejný jako SSH).
- **Výhody:** Nabízí vyšší zabezpečení než FTP díky šifrování veškeré komunikace a používá pouze jedno spojení pro přenos i příkazy.

D) SCP (Secure Copy Protocol)

- **Účel:** Protokol pro bezpečné kopírování souborů mezi počítači přes SSH.
- **Port:** Používá port 22 (stejně jako SFTP a SSH).
- **Vlastnosti:** Podobně jako SFTP, využívá šifrování, ale je méně flexibilní, protože umožňuje pouze jednoduché kopírování souborů.

E) SMB (Server Message Block) a CIFS (Common Internet File System)

- **Účel:** Protokoly pro sdílení souborů v rámci lokální sítě (typicky v prostředí Microsoft Windows).
- **Porty:** Používají porty 137–139 (NetBIOS) a 445 (SMB přímé připojení).
- **Vlastnosti:** Umožňuje sdílení souborů, tiskáren a dalších zdrojů v rámci sítě. SMB může být zabezpečen pomocí NTLM nebo Kerberos autentizace.

F) NFS (Network File System)

- **Účel:** Protokol pro sdílení souborů v síti, obvykle mezi systémy Linux a Unix.

- **Porty:** Různé porty (obvykle 2049) podle nastavení serveru.
- **Vlastnosti:** Umožňuje připojení vzdálených souborových systémů, vhodné pro sdílení dat mezi Unixovými a Linuxovými servery.

G) HTTP/HTTPS

- **Účel:** Protokoly pro přenos souborů přes webový prohlížeč.
- **Porty:** 80 (HTTP) a 443 (HTTPS).
- **Vlastnosti:** Vhodné pro stahování a nahrávání souborů přes webové rozhraní. HTTPS zajišťuje šifrování dat při přenosu.

11.3. Zabezpečení při přenosu souborů

Při sdílení souborů a přenosu dat je zásadní dbát na bezpečnost, aby nedošlo k úniku dat nebo neoprávněnému přístupu.
A) Šifrování dat

- **FTPS a SFTP** zajišťují šifrování dat během přenosu pomocí SSL/TLS nebo SSH, čímž chrání přihlašovací údaje a obsah před odposlechem.
- **HTTPS** zajišťuje šifrování při přenosu přes webové rozhraní.

B) Autentizace

- **Autentizace uživatelů:** Použití uživatelského jména a hesla pro přístup k serveru.
- **Dvoufaktorová autentizace (2FA):** Další bezpečnostní vrstva, kde se kromě hesla vyžaduje jednorázový kód (např. přes SMS nebo aplikaci).
- **Certifikáty:** Použití certifikátů pro ověřování totožnosti serveru nebo klienta (FTPS s klientskými certifikáty).

C) Ochrana proti útokům

- **Firewall a pravidla:** Konfigurace firewallu pro omezení přístupu k portům FTP, SFTP a FTPS pouze z důvěryhodných IP adres.
- **Chráněný přístup k portům:** Použití pasivního režimu u FTP, kdy server určí bezpečný port, který může být otevřen pro příchozí připojení.
- **Blokování IP adres:** Ochrana proti brute force útokům tím, že server blokuje IP adresy po opakovaných neúspěšných přihlášeních.

D) Nastavení oprávnění

- **Omezení přístupu k souborům:** Nastavení uživatelských práv pro čtení, zápis a mazání souborů.
- **Oddělení účtů:** Vytvoření účtů s omezenými právy, například pro anonymní uživatele, aby měli pouze přístup ke čtení.

E) Monitorování a protokolování

- **Logování přístupu:** Zaznamenávání všech přístupů a aktivit uživatelů na serveru, což umožňuje auditování a odhalení podezřelé aktivity.
- **Automatické upozornění:** Upozornění administrátora v případě pokusu o neoprávněný přístup nebo jinou neobvyklou aktivitu.

F) Aktualizace softwaru

- **Pravidelné aktualizace:** Zajišťují, že software (servery, klienty) obsahuje nejnovější bezpečnostní záplaty a minimalizuje zranitelnosti.

11.4. Výhody a nevýhody protokolů pro sdílení souborů

Protokol	Výhody	Nevýhody
FTP	Jednoduchost, široká podpora	Nešifrovaný, bezpečnostní riziko
FTPS	Šifrovaný přenos (SSL/TLS)	Složitější konfigurace, vyžaduje více portů
SFTP	Vysoké zabezpečení, jedno spojení (port 22)	Vyšší náročnost na výkon
SCP	Jednoduchý, bezpečný přenos	Omezené možnosti správy a přístupu
SMB	Dobrá podpora pro Windows, sdílení souborů a tiskáren	Méně bezpečný v nezabezpečených sítích
NFS	Vhodné pro Unix/Linux sítě, sdílení systému souborů	Omezená podpora pro jiné platformy

Protokol	Výhody	Nevýhody
HTTPS	Šifrovaný přenos přes web	Nevhodný pro velké objemy souborů

11.5. Závěr

FTP a jeho zabezpečené verze, FTPS a SFTP, jsou základními protokoly pro přenos souborů mezi klienty a servery. Zabezpečení přenosu souborů je nezbytné pro ochranu citlivých dat, a to zejména pomocí šifrování, autentizace a monitorování. Každý protokol má své výhody a nevýhody, proto je výběr protokolu závislý na specifických požadavcích, jako jsou bezpečnost, výkon a jednoduchost použití. Pro přenosy v nezabezpečeném prostředí je důležité používat šifrované verze protokolů, aby nedošlo k úniku dat a zneužití přihlašovacích údajů.

12. Služba WWW

Princip, protokoly, zabezpečení

12.1. Princip služby WWW (World Wide Web)

Služba WWW (World Wide Web) je systém, který umožňuje přístup k hypertextovým dokumentům propojeným odkazy a uloženým na serverech po celém světě. WWW je postaveno na protokolech, které definují, jakým způsobem se přenáší data mezi webovými servery a klienty (webovými prohlížeči).
Uživatelé používají prohlížeče k prohlížení webových stránek, které jsou psány v jazycích jako HTML (Hypertext Markup Language). Webové stránky mohou obsahovat text, obrázky, multimédia a další interaktivní prvky.
Princip fungování WWW

1. **Vytvoření požadavku:** Uživatel zadá URL adresu nebo klikne na odkaz. Prohlížeč vytvoří požadavek na webový server.
2. **Komunikace s DNS:** Prohlížeč zjistí IP adresu serveru, na kterém je požadovaná stránka, pomocí DNS (Domain Name System).
3. **Připojení k serveru:** Prohlížeč naváže spojení se serverem pomocí HTTP nebo HTTPS.
4. **Odeslání požadavku:** Prohlížeč pošle na server požadavek na konkrétní stránku nebo zdroje.
5. **Zpracování a odpověď:** Server odpoví zasláním HTML kódu a dalších zdrojů, které tvoří stránku.
6. **Zobrazení stránky:** Prohlížeč zpracuje HTML, CSS, JavaScript a další zdroje a zobrazí webovou stránku uživateli.

12.2. Protokoly služby WWW

WWW používá několik klíčových protokolů pro komunikaci a přenos dat mezi servery a klienty. Hlavní protokoly jsou **HTTP** a **HTTPS**.
A) HTTP (Hypertext Transfer Protocol)

- **Účel:** HTTP je základní protokol pro přenos dat na WWW, který umožňuje prohlížečům komunikovat se servery a načítat webové stránky.
- **Port:** Standardně používá port 80.
- **Princip fungování:** HTTP funguje na principu požadavků a odpovědí – prohlížeč (klient) pošle na server požadavek a server odpoví obsahem.
- **Bezestavový protokol:** HTTP je bezestavový protokol, což znamená, že každý požadavek je nezávislý na předchozím. Webové aplikace proto používají technologie jako cookies nebo session ID pro uchování stavu.
- Verze:
    - **HTTP/1.0:** První verze používala jedno spojení na každý požadavek.
    - **HTTP/1.1:** Vylepšená verze umožňuje uchovávat spojení (keep-alive), což snižuje zátěž serveru a zlepšuje rychlost načítání stránek.
    - **HTTP/2:** Přidává multiplexing, kompresi hlaviček a prioritizaci požadavků, čímž zvyšuje výkon a efektivitu.
    - **HTTP/3:** Nejnovější verze využívající QUIC protokol, který přináší nižší latenci a větší bezpečnost.

B) HTTPS (HTTP Secure)

- **Účel:** HTTPS je zabezpečená verze HTTP, která přidává šifrování komunikace pomocí protokolu TLS (Transport Layer Security).
- **Port:** Standardně používá port 443.
- **Šifrování:** HTTPS zajišťuje, že data přenášená mezi klientem a serverem jsou šifrována, čímž chrání citlivé informace před odposlechem a útoky typu „man-in-the-middle“.
- **Autentizace:** HTTPS používá certifikáty (SSL/TLS certifikáty), které ověřují totožnost serveru a zajišťují, že komunikace je bezpečná.
- **Výhody:** HTTPS poskytuje důvěrnost a integritu dat a je nezbytný pro zabezpečení moderních webových aplikací, zejména u e-commerce a bankovních webů.

C) Protokoly podporující HTTP a HTTPS

- **TLS (Transport Layer Security):** Zajišťuje šifrování a bezpečnost pro HTTPS. Je následníkem SSL a poskytuje bezpečný přenos dat.
- **DNS:** Používá se k překladu doménových jmen na IP adresy. DNS je nezbytné pro vyhledávání serverů, na kterých jsou webové stránky hostovány.

12.3. Zabezpečení služby WWW

Webová komunikace může být zranitelná vůči různým typům útoků, a proto se používají různé bezpečnostní techniky a protokoly, aby se zajistilo bezpečí uživatelů i serverů.

A) Šifrování pomocí TLS/SSL

- **TLS/SSL certifikáty:** Certifikáty, které zajišťují šifrování a ověření identity serveru. Tyto certifikáty vydávají certifikační autority (CA), které potvrzují pravost serveru.
- **Zabránění odposlechu:** Šifrování dat mezi klientem a serverem zajišťuje, že třetí strany nemohou číst přenášené informace.

B) Autentizace a certifikáty

- **DV (Domain Validation):** Certifikát, který ověřuje vlastnictví domény.
- **OV (Organization Validation):** Certifikát, který ověřuje identitu organizace vlastnící doménu.
- **EV (Extended Validation):** Certifikát s nejvyšší úrovní ověření, který poskytuje vizuální indikaci (např. zelený zámeček), že web patří důvěryhodné organizaci.

C) Ochrana proti útokům

- **HTTP Strict Transport Security (HSTS):** Ochranný mechanismus, který informuje prohlížeče, aby vždy připojovaly uživatele přes HTTPS, čímž se zabrání přístupu přes nezabezpečené HTTP.
- **Zabezpečení proti XSS (Cross-Site Scripting):** Ochrana proti útokům, při kterých útočníci vkládají škodlivý kód do webových aplikací, aby získali přístup k citlivým datům.
- **CSRF (Cross-Site Request Forgery):** Ochrana proti útokům, při kterých útočník podstrčí uživateli škodlivý požadavek, aby přiměl aplikaci vykonat nežádoucí akci.
- **Content Security Policy (CSP):** Mechanismus, který zabraňuje vkládání škodlivého obsahu do webových aplikací tím, že definuje povolené zdroje obsahu.

D) Bezpečnostní hlavičky HTTP

- **Content-Security-Policy (CSP):** Definuje, jaké zdroje obsahu (např. skripty, styly, obrázky) může webová stránka načítat. Pomáhá chránit proti XSS.
- **X-Frame-Options:** Zabraňuje vložení webu do rámce, což chrání proti clickjacking útokům.
- **X-XSS-Protection:** Aktivuje ochranu proti XSS útokům v prohlížečích.
- **X-Content-Type-Options:** Zabraňuje prohlížečům interpretovat soubory jiného typu, než je deklarováno.

E) Ověřování a správa session

- **Cookies:** Slouží k ukládání identifikačních údajů (např. session ID) na straně klienta. Cookies by měly být nastaveny s atributy Secure (přenos pouze přes HTTPS) a HttpOnly (pouze pro HTTP požadavky).
- **Session management:** Řízení relací mezi uživatelem a serverem, které zajišťuje, že session ID nelze snadno odcizit nebo zneužít.

12.4. Výhody a nevýhody služby WWW

Výhody

- **Dostupnost a globální přístup:** WWW umožňuje přístup k informacím odkudkoliv a kdykoliv, pokud je k dispozici internetové připojení.
- **Rychlost a efektivita:** S informacemi dostupnými online je komunikace mezi uživateli a přístup k informacím rychlý.
- **Multimediální obsah:** WWW podporuje text, obrázky, videa a další formy obsahu, což zlepšuje uživatelský zážitek.
- **Interaktivita:** Moderní webové stránky umožňují interakci s uživateli, například prostřednictvím formulářů, komentářů a sociálních médií.

Nevýhody

- **Bezpečnostní rizika:** WWW je zranitelné vůči různým typům kybernetických útoků, jako je phishing, malware a útoky typu „man-in-the-middle“.
- **Závislost na internetovém připojení:** WWW je závislé na připojení k internetu, což může být omezením v oblastech s horším připojením.
- **Sledování a soukromí:** Online sledování a shromažďování dat mohou vést ke ztrátě soukromí uživatelů.
- **Informační přetížení:** Na internetu je obrovské množství informací, které mohou být matoucí nebo neověřené.

12.5. Příklady chybových stavů ve službě WWW

Při komunikaci mezi klientem a serverem v rámci služby WWW mohou nastat různé chybové stavy, které se označují jako **HTTP stavové kódy**. Tyto kódy jsou třímístná čísla, kde první číslice označuje typ odpovědi. Nejčastější chybové stavy mají čísla začínající číslicemi 4 (chyby na straně klienta) a 5 (chyby na straně serveru).
Chybové stavy na straně klienta (kódy 4xx)

1. 400 Bad Request
    - **Význam:** Server nerozumí požadavku kvůli chybnému syntaxi. Například může jít o neplatnou URL nebo chybně formátovaná data.
    - **Příklad:** Klient odešle požadavek s neplatnou strukturou nebo chybně zakódovanými parametry.
2. 401 Unauthorized
    - **Význam:** Přístup k požadovanému zdroji je omezen a vyžaduje autentizaci. Chyba nastane, když není poskytnut správný autentizační token nebo chybí oprávnění.
    - **Příklad:** Uživatel se pokusí přistoupit na chráněnou stránku bez přihlášení.
3. 403 Forbidden
    - **Význam:** Server rozumí požadavku, ale odmítá ho zpracovat. Uživatel nemá oprávnění pro přístup k požadovanému zdroji.
    - **Příklad:** Uživatel se pokusí přistoupit na stránku, ke které nemá oprávnění, např. administrativní rozhraní.
4. 404 Not Found
    - **Význam:** Požadovaný zdroj nebyl nalezen. Tato chyba znamená, že URL neodpovídá žádnému existujícímu obsahu na serveru.
    - **Příklad:** Uživatel zadá špatnou URL nebo pokusí se přistoupit k obsahu, který byl odstraněn.
5. 408 Request Timeout
    - **Význam:** Server čekal příliš dlouho na dokončení požadavku od klienta a spojení vypršelo.
    - **Příklad:** Klient se připojí k serveru, ale neodešle žádná data, což vede k ukončení spojení kvůli časovému limitu.

Chybové stavy na straně serveru (kódy 5xx)

1. 500 Internal Server Error
    - **Význam:** Server narazil na neočekávanou chybu nebo selhání, které mu zabránilo dokončit požadavek.
    - **Příklad:** Nastane při selhání skriptu, špatné konfiguraci serveru nebo jiné interní chybě.
2. 501 Not Implemented
    - **Význam:** Server nepodporuje metodu požadavku, který klient použil.
    - **Příklad:** Server nemusí podporovat novější HTTP metody nebo požadavky, které nejsou implementovány.
3. 502 Bad Gateway
    - **Význam:** Server, který funguje jako proxy nebo brána, dostal neplatnou odpověď od jiného serveru.
    - **Příklad:** Chyba může nastat, pokud proxy server nedostane platnou odpověď od backendového serveru nebo pokud dojde k dočasnému výpadku.
4. 503 Service Unavailable
    - **Význam:** Server je dočasně nedostupný, obvykle kvůli údržbě nebo přetížení. Chyba může signalizovat, že služba bude obnovena později.
    - **Příklad:** Server je přetížený velkým množstvím požadavků nebo probíhá plánovaná údržba.
5. 504 Gateway Timeout
    - **Význam:** Server, který funguje jako brána nebo proxy, neobdržel včas odpověď od jiného serveru.
    - **Příklad:** Nastane, pokud proxy nebo load balancer čeká na odpověď z backendu, která však nepřijde kvůli vypršení časového limitu.
6. 505 HTTP Version Not Supported
    - **Význam:** Server nepodporuje verzi protokolu HTTP, kterou klient požaduje.
    - **Příklad:** Klient se pokusí přistoupit k serveru s verzí HTTP, kterou server nezvládá, například velmi stará nebo experimentální verze HTTP.

12.6. Závěr

Služba WWW je klíčovým pilířem internetu, který umožňuje snadný přístup k informacím prostřednictvím webových stránek a hypertextových odkazů. WWW je založeno na protokolech HTTP a HTTPS, přičemž HTTPS zajišťuje šifrování a zabezpečení dat. K ochraně dat a uživatelů před bezpečnostními hrozbami se využívají různé metody, včetně TLS, bezpečnostních

hlaviček HTTP, HSTS a správného řízení session a cookies. Zabezpečení služby WWW je klíčové pro důvěru uživatelů a ochranu před rostoucími kybernetickými hrozbami.
Chybové stavy v rámci služby WWW jsou důležitou součástí komunikace mezi klientem a serverem. Tyto stavové kódy pomáhají uživatelům a administrátorům rychle identifikovat problémy a opravit je. Zatímco chyby na straně klienta (4xx) se často týkají nesprávně vytvořených požadavků, chyby na straně serveru (5xx) mohou vyžadovat zásah do konfigurace serveru nebo jeho infrastruktury.

13. Zabezpečení sítí

Druhy hrozeb, firewall, VPN

13.1. Druhy hrozeb pro síťovou bezpečnost

Síťová bezpečnost čelí různým hrozbám, které mohou mít za následek narušení důvěrnosti, integrity nebo dostupnosti dat. Hrozby se dělí na různé typy v závislosti na způsobu útoku a cíli útočníka.

A) Malware (škodlivý software)

- **Popis:** Škodlivý software, který je navržen tak, aby infikoval systémy, narušoval jejich provoz, kradl data nebo umožňoval přístup útočníkovi.
- Příklady:
  - **Viry:** Šíří se infikováním jiných souborů.
  - **Trojské koně:** Vydávají se za legitimní software a po instalaci umožňují přístup útočníkovi.
  - **Ransomware:** Šifruje data a vyžaduje výkupné za jejich odemknutí.

B) Phishing

- **Popis:** Sociotechnická technika, při které útočník předstírá důvěryhodný subjekt a podvádí uživatele, aby poskytl citlivé údaje, jako jsou přihlašovací údaje nebo finanční informace.
- **Formy:** E-maily, SMS nebo webové stránky, které vypadají jako legitimní.

C) Man-in-the-Middle (MITM) útoky

- **Popis:** Útočník se „vmísí“ mezi komunikaci dvou stran a získává nebo mění přenášená data, aniž by o tom strany věděly.
- **Příklady:** Útoky na veřejné Wi-Fi sítě, kde útočník monitoruje nebo upravuje komunikaci.

D) Distributed Denial of Service (DDoS)

- **Popis:** Útok zaměřený na přetížení serveru nebo sítě, což způsobí nedostupnost služby pro legitimní uživatele.
- **Metody:** Útoky s velkým množstvím požadavků (např. pingování), botnety (síť infikovaných zařízení).

E) SQL Injection (SQLi)

- **Popis:** Útok, při kterém útočník vkládá škodlivý kód do SQL dotazu, což mu umožňuje přístup k databázi a získání nebo úpravu dat.
- **Cíl:** Často útoky na webové aplikace, které nesprávně zpracovávají vstupy.

F) Cross-Site Scripting (XSS)

- **Popis:** Útok, kdy útočník vkládá škodlivý JavaScript do webových stránek, což mu umožňuje získat data uživatele nebo provádět akce jeho jménem.
- **Cíl:** Zranitelné webové aplikace, kde je možné vložit škodlivý obsah do webového rozhraní.

G) Insider Threats (vnitřní hrozby)

- **Popis:** Hrozby pocházející od zaměstnanců nebo důvěryhodných osob, kteří mají přístup k síti. Tyto osoby mohou zneužít svůj přístup pro osobní zisk nebo na příkaz třetí strany.
- **Příklady:** Krádež dat, sabotáž nebo úmyslné šíření škodlivého softwaru.

13.2. Firewall

Firewall je klíčové zařízení nebo software, které chrání síť tím, že kontroluje a filtruje příchozí a odchozí síťový provoz podle předem definovaných pravidel. Firewall zajišťuje, aby neoprávněný provoz neměl přístup do sítě nebo konkrétních zařízení.

Typy firewallů

1. Packet Filtering Firewall (Filtrační firewall)
   - **Popis:** První generace firewallů, která analyzuje jednotlivé pakety (části dat) na základě IP adres, portů a protokolů.
   - **Výhody:** Rychlý a jednoduchý na konfiguraci, ale omezený pouze na základní informace v paketu.
   - **Nevýhody:** Není schopen analyzovat obsah dat, pouze základní síťové informace, což může vést k průniku sofistikovanějšími útoky.
2. Stateful Inspection Firewall (Stavový firewall)

- **Popis:** Udržuje stavovou tabulku spojení a sleduje stav každého spojení, což umožňuje rozhodovat na základě předchozí komunikace.
- **Výhody:** Větší bezpečnost než u packet filtering firewallů, jelikož udržuje přehled o průběhu spojení.
- **Nevýhody:** Vyšší náročnost na výkon než u základních packet filtering firewallů.

3. Application Layer Firewall (Aplikační firewall)
   - **Popis:** Analyzuje data na aplikační vrstvě (např. HTTP, FTP) a může kontrolovat obsah konkrétních požadavků nebo odpovědí.
   - **Výhody:** Schopnost rozeznat a blokovat specifické hrozby v rámci aplikací.
   - **Nevýhody:** Vyšší zátěž na systém a může zpomalovat přenos dat.
4. Next-Generation Firewall (NGFW)
   - **Popis:** Kombinuje vlastnosti stavových a aplikačních firewallů a zahrnuje další pokročilé bezpečnostní funkce, jako je detekce hrozeb, kontrola přístupu na základě uživatelů a inspekce SSL/TLS.
   - **Výhody:** Vysoká úroveň zabezpečení s možností integrace s dalšími bezpečnostními systémy.
   - **Nevýhody:** Vyšší cena a náročnost na konfiguraci i údržbu.
5. Proxy Firewall
   - **Popis:** Přeposílá provoz mezi klientem a serverem a skrývá identitu skutečného klienta. Proxy firewall může kontrolovat a filtrovat obsah.
   - **Výhody:** Chrání skutečnou IP adresu uživatele, snižuje možnost přímého útoku na klienta.
   - **Nevýhody:** Může zpomalit komunikaci a je náročnější na konfiguraci.

Funkce firewallu

- **Filtrování paketů:** Kontrola jednotlivých paketů a blokování na základě předdefinovaných pravidel.
- **Detekce a prevence hrozeb:** Firewall NGFW může detekovat a zastavit známé útoky, jako jsou malware, DDoS útoky a jiné hrozby.
- **Inspekce šifrovaného provozu:** Analyzuje šifrovanou komunikaci pomocí TLS/SSL, aby detekoval hrozby skryté uvnitř šifrovaných dat.

13.3. 3. VPN (Virtual Private Network)

VPN (virtuální privátní síť) je technologie, která umožňuje bezpečné připojení k síti nebo zařízení prostřednictvím šifrovaného „tunelu“ přes veřejnou nebo nedůvěryhodnou síť, jako je internet. VPN chrání data, která jsou přenášena, a umožňuje uživatelům přístup k síti, jako by byli fyzicky připojeni.
Princip fungování VPN

1. **Zahájení spojení:** Uživatel se připojí k VPN serveru prostřednictvím klientské aplikace, která šifruje data na straně klienta.
2. **Šifrovaný tunel:** VPN klient vytvoří šifrovaný tunel přes veřejnou síť, přes který probíhá bezpečná komunikace s VPN serverem.
3. **Přístup k síti:** VPN server dešifruje data a předá je do cílové sítě. Pro vzdáleného uživatele to vypadá, jako by byl připojen přímo k síti organizace.

Typy VPN

1. Remote Access VPN (VPN pro vzdálený přístup)
   - **Popis:** Umožňuje jednotlivým uživatelům přístup k podnikové síti odkudkoliv. Obvykle se používá pro zaměstnance, kteří potřebují vzdálený přístup k firemním zdrojům.
   - **Příklady použití:** Práce z domova, přístup k interním aplikacím a datům.
2. Site-to-Site VPN (VPN pro propojení poboček)
   - **Popis:** Spojuje dvě nebo více sítí (např. pobočky jedné firmy) a umožňuje jejich komunikaci přes internet. Umožňuje bezpečnou komunikaci mezi geograficky oddělenými lokacemi.
   - **Příklady použití:** Spojení poboček firmy, která má kanceláře na více místech.
3. Client-to-Site VPN
   - **Popis:** Kombinace dvou výše uvedených typů, umožňuje připojení uživatelů i celých sítí k hlavní síti organizace.

Protokoly pro VPN

1. PPTP (Point-to-Point Tunneling Protocol)
   - **Výhody:** Jednoduchá konfigurace a široká podpora.
   - **Nevýhody:** Nízká úroveň zabezpečení, používá zastaralé šifrování.

2. L2TP/IPsec (Layer 2 Tunneling Protocol/Internet Protocol Security)
    - **Výhody:** Dobrá úroveň zabezpečení díky IPsec, kombinuje šifrování a autentizaci.
    - **Nevýhody:** Vyšší náročnost na konfiguraci a výkon.
3. OpenVPN
    - **Výhody:** Otevřený protokol s vysokou úrovní bezpečnosti, podporuje silné šifrování a je velmi flexibilní.
    - **Nevýhody:** Vyžaduje instalaci klienta a je náročnější na konfiguraci než PPTP nebo L2TP.
4. IKEv2/IPsec (Internet Key Exchange v2)
    - **Výhody:** Vysoká stabilita, rychlá rekonfigurace při změně sítě, což je vhodné pro mobilní zařízení.
    - **Nevýhody:** Vyžaduje podporu na obou stranách (klient a server) a složitější konfigurace.

13.4. Závěr

Zabezpečení sítě zahrnuje ochranu před různými typy hrozeb, včetně malwaru, phishingu, MITM útoků a dalších. Firewall a VPN jsou klíčovými prvky pro zajištění bezpečnosti sítě. Firewally pomáhají filtrovat provoz a chránit síť před útoky, zatímco VPN zajišťuje bezpečný a šifrovaný přístup k síti přes veřejnou síť, což umožňuje bezpečnou komunikaci na dálku.

14. Diagnostika sítě

Nástroje pro analýzu síťového HW a síťové komunikace, postup odstraňování závad

14.1. Diagnostika sítě – úvod

Diagnostika sítě je proces identifikace a řešení problémů v síťové infrastruktuře. Cílem je odhalit problémy, které brání plynulému přenosu dat nebo narušují provoz sítě, ať už na úrovni hardwaru, protokolů nebo aplikací.

14.2. Nástroje pro analýzu síťového hardwaru

Síťový hardware zahrnuje zařízení, jako jsou routery, switche, servery a kabeláž. Diagnostické nástroje pro hardware se zaměřují na monitorování provozu a kontrolu správné funkce fyzických komponent sítě.

A) Multimetr

- **Použití:** Měří napětí, proud a odpor v elektrických obvodech.
- **Využití v síťové diagnostice:** Ověřuje funkčnost kabeláže, napájení zařízení a pomáhá lokalizovat fyzické závady v síti.

B) Kabelový tester

- **Účel:** Ověřuje funkčnost kabelů a konektorů.
- **Funkce:** Identifikuje problémy s připojením, jako jsou špatně zapojené vodiče, přerušené kabely nebo nesprávné propojení párů.
- **Výhoda:** Rychlá kontrola, zda kabely nejsou poškozené nebo špatně zakončené.

C) Toner a sondy (cable tracer)

- **Účel:** Pomáhá vystopovat konkrétní kabel mezi množstvím kabelů ve svazku.
- **Využití:** Sonda generuje zvukový signál, který usnadňuje lokalizaci kabelu a určení jeho konců.

D) Síťový analyzátor (Network Analyzer)

- **Účel:** Monitoruje a analyzuje provoz v síti.
- **Funkce:** Měří výkon, přenosové rychlosti a latenci v síti a identifikuje zatížené úseky.
- **Využití:** Pomáhá identifikovat problémové oblasti a analyzovat celkový výkon síťového hardwaru.

E) Loopback tester

- **Účel:** Kontroluje porty a kabely na síťových zařízeních tím, že simuluje uzavřený obvod.
- **Funkce:** Testuje správnou funkci síťového portu a jeho schopnost odesílat a přijímat data.
- **Využití:** Rychlá diagnostika funkčnosti jednotlivých portů na routerech, switchích a síťových kartách.

14.3. Nástroje pro analýzu síťové komunikace

Tyto nástroje se zaměřují na analýzu přenosu dat v síti, kontrolu protokolů a sledování síťového provozu.

A) Ping

- **Účel:** Základní nástroj pro ověření konektivity mezi dvěma zařízeními v síti.
- **Princip:** Odesílá ICMP požadavky na zvolenou IP adresu a měří dobu odezvy.
- **Využití:** Kontrola dostupnosti síťových zařízení, měření latence a identifikace přerušení spojení.

B) Traceroute (Windows: tracert)

- **Účel:** Sleduje trasu, kterou paket prochází přes jednotlivé routery k cílové IP adrese.
- **Princip:** Zobrazuje každý router, kterým paket prochází, a dobu odezvy.
- **Využití:** Diagnostika problémů na konkrétních úsecích sítě, detekce pomalých nebo nedostupných uzlů.

C) Netstat

- **Účel:** Zobrazuje aktivní síťová spojení a otevřené porty na zařízení.
- **Princip:** Zobrazuje informace o aktuálních spojení, IP adresách a portech, protokolech a stavu spojení.
- **Využití:** Diagnostika síťových služeb, identifikace neobvyklých nebo podezřelých spojení.

D) Nslookup

- **Účel:** Nástroj pro diagnostiku a testování DNS serverů a jejich záznamů.
- **Princip:** Umožňuje zjistit IP adresu doménového jména nebo doménové jméno IP adresy.
- **Využití:** Kontrola správné funkce DNS serveru, řešení problémů s překladem jmen.

E) Wireshark

- **Účel:** Pokročilý nástroj pro analýzu síťového provozu na paketové úrovni.
- **Princip:** Zachycuje a zobrazuje detaily o jednotlivých paketech, včetně protokolů, IP adres, portů a obsahu.
- **Využití:** Diagnostika složitých problémů s komunikací, analýza bezpečnostních hrozeb a sledování specifických přenosů.

F) ipconfig (Windows) a ifconfig/ip (Linux)

- **Účel:** Zobrazuje a konfiguruje síťová rozhraní na zařízení.
- **Funkce:** Ukazuje IP adresu, masku podsítě, výchozí bránu a stav síťového adaptéru.
- **Využití:** Rychlá kontrola konfigurace síťového rozhraní, řešení problémů s připojením.

G) ARP (Address Resolution Protocol)

- **Účel:** Zobrazuje a spravuje ARP tabulku, která obsahuje mapování IP adres na MAC adresy.
- **Využití:** Diagnostika a odstraňování problémů s komunikací na úrovni linkové vrstvy, identifikace neplatných nebo nesprávně mapovaných adres.

H) Nmap (Network Mapper)

- **Účel:** Nástroj pro skenování sítě a detekci aktivních zařízení, otevřených portů a běžících služeb.
- **Využití:** Zabezpečení a audit sítě, detekce neautorizovaných zařízení nebo služeb, kontrola správného nastavení firewallu.

14.4. Postup odstraňování závad v síti

Odstraňování závad je systematický proces identifikace a odstranění příčin problémů v síti. Zde je standardní postup kroků při diagnostice a řešení síťových problémů.

Krok 1: Identifikace problému

- **Popis problému:** Zjistěte, co přesně uživatelé hlásí (např. pomalé připojení, ztráta spojení, nemožnost připojení k serveru).
- **Dotazování:** Položte uživatelům doplňující otázky a zkuste zjistit, kdy problém začal, zda ovlivňuje jedno nebo více zařízení.

Krok 2: Testování konektivity

- **Ping:** Ověřte připojení pomocí ping příkazu, abyste zjistili, zda je cíl dostupný.
- **Traceroute:** Zkontrolujte, zda data prochází přes správné routery a zda nejsou v trase zpoždění nebo výpadky.
- **Nslookup:** Ověřte funkci DNS, pokud je problém s překladem jmen.

Krok 3: Diagnostika hardwaru

- **Kontrola kabelů:** Použijte kabelový tester nebo multimetr k ověření správného připojení a funkčnosti kabeláže.
- **Zkouška portů:** Pokud se jedná o problém s konkrétním zařízením, otestujte porty na switchi nebo routeru.
- **Loopback test:** Proveďte loopback test na síťovém portu zařízení, abyste ověřili, zda je port funkční.

Krok 4: Kontrola síťové konfigurace

- **ipconfig/ifconfig:** Zkontrolujte IP adresu, masku podsítě a výchozí bránu na problémovém zařízení.
- **ARP tabulka:** Zkontrolujte ARP tabulku pro záznamy IP/MAC adres, které mohou být nesprávně mapované.
- **Nastavení firewallu a směrování:** Zkontrolujte, zda není firewall nebo směrovací tabulka špatně nakonfigurovaná a nezabraňuje připojení.

Krok 5: Analýza síťového provozu

- **Wireshark:** Zachyťte a analyzujte síťový provoz na problémovém zařízení nebo mezi konkrétními body, abyste zjistili, zda nedochází k ztrátě paketů nebo přenosu neobvyklých dat.
- **Netstat a Nmap:** Zkontrolujte otevřené porty a aktivní spojení na síťovém zařízení pro identifikaci neobvyklých nebo neoprávněných spojení.

Krok 6: Testování a obnovení připojení

- **Reset zařízení:** Pokud není příčina stále zřejmá, restartujte zařízení (routery, switche, koncová zařízení), což může obnovit spojení nebo vyřešit dočasné chyby.
- **Ověření připojení:** Po změnách a opravách otestujte připojení pomocí příkazů ping, traceroute nebo dalších nástrojů.

Krok 7: Dokumentace a sledování

- **Záznam změn:** Zaznamenejte provedené kroky, zjištění a opravy, které vedly k vyřešení problému.
- **Monitorování:** Nastavte monitoring, aby se zabránilo opakování problému. Monitorujte výkon sítě a případně upravte konfiguraci zařízení podle potřeby.

14.5. Závěr

Diagnostika sítě a odstraňování závad vyžaduje kombinaci hardwarových a softwarových nástrojů. Klíčové je postupovat systematicky od identifikace problému přes kontrolu hardwaru a konfigurace až po analýzu síťové komunikace. Důležitá je dokumentace postupu a sledování stavu sítě, aby se podobným problémům předešlo v budoucnu.

Operační systémy

15. Architektura operačních systémů

základní pojmy, funkce OS, typy OS, typy architektur OS, architektura Windows, architektura Linux

15.1. Základní pojmy

- **Operační systém (OS)**: Software, který slouží jako prostředník mezi hardwarem a uživatelem a umožňuje správu zdrojů počítače (např. procesoru, paměti, úložiště). OS poskytuje prostředí pro spouštění aplikací a zajišťuje jejich bezpečnost a stabilitu.
- **Jádro (kernel)**: Základní součást operačního systému, která řídí přístup k hardwaru a zajišťuje klíčové služby, jako je správa paměti, plánování procesů a správa vstupu a výstupu.
- **Uživatelské rozhraní (UI)**: Prostředí, které umožňuje interakci mezi uživatelem a operačním systémem. UI může být grafické (GUI) nebo textové (CLI – Command Line Interface).
- **Proces a vlákno**: Proces je instance spuštěného programu, zatímco vlákno (thread) je menší jednotka vykonávání, která může být spravována v rámci procesu.
- **Správa paměti**: Mechanismus, který OS využívá k efektivnímu přidělování a sledování paměťových zdrojů.

15.2. Funkce operačního systému

Operační systém plní různé základní funkce, které zajišťují, že počítačový systém funguje správně a efektivně:

1. **Správa procesů**: Zajišťuje tvorbu, plánování a ukončování procesů a řídí jejich přístup k procesoru. Obsahuje mechanismy, jako je multitasking a plánování (scheduling), které umožňují více procesům běžet současně.
2. **Správa paměti**: OS spravuje fyzickou a virtuální paměť, přiděluje paměť procesům a udržuje jejich izolaci, čímž zajišťuje, že procesy nebudou zasahovat do paměťových prostorů jiných procesů.
3. **Správa vstupu a výstupu (I/O)**: OS řídí interakci s periferními zařízeními, jako jsou disky, tiskárny nebo síťové karty. Umožňuje komunikaci mezi hardwarem a softwarem pomocí ovladačů.
4. **Správa souborů**: OS spravuje souborový systém, který umožňuje ukládání, organizaci a zabezpečení dat na úložišti.
5. **Bezpečnost a ochrana**: Zajišťuje kontrolu přístupu k datům a zdrojům systému, poskytuje zabezpečení proti neoprávněnému přístupu, šifrování dat a ochranu proti škodlivému softwaru.
6. **Uživatelské rozhraní**: Zajišťuje prostředí pro interakci mezi uživatelem a systémem. Rozhraní může být grafické (GUI) nebo příkazové (CLI).

15.3. Typy operačních systémů

Operační systémy lze rozdělit na základě jejich zaměření a funkcionality:

1. Jednouživatelské a víceuživatelské OS:
   - **Jednouživatelské OS**: Podporují pouze jednoho uživatele v daném okamžiku, např. starší verze MS-DOS.
   - **Víceuživatelské OS**: Umožňují více uživatelům pracovat současně, např. Unix nebo Windows Server.
2. Jednoúlohové a víceúlohové OS:
   - **Jednoúlohové OS**: Mohou spouštět pouze jednu úlohu současně, např. MS-DOS.
   - **Víceúlohové OS**: Podporují multitasking, kde mohou běžet více procesů současně, např. Linux a moderní verze Windows.
3. **Síťové OS**: Navrženy pro správu a sdílení síťových zdrojů, jako jsou soubory a tiskárny. Příkladem je Novell NetWare a Windows Server.
4. **Distribuované OS**: Umožňují zpracování dat na více propojených počítačích, které spolupracují jako jeden systém.
5. **Vestavěné OS (Embedded OS)**: Optimalizovány pro konkrétní zařízení nebo aplikace, např. OS na routerech, chytrých telefonech (Android, iOS).
6. **Mobilní OS**: Specifické OS pro mobilní zařízení, např. Android a iOS.

15.4. Typy architektur OS

Architektura OS definuje, jak je organizováno jádro a komponenty operačního systému.

1. Monolitické jádro:

- Všechny základní služby OS běží v jedné velké struktuře v jádře. Je rychlé a efektivní, ale pokud dojde k chybě, celý systém se může zhroutit.
- **Příklad:** Linux, Unix.

2. Mikrojádro:
   - Rozděluje jádro na menší moduly, které běží nezávisle. Snižuje riziko pádů systému, ale komunikace mezi moduly může zpomalit výkon.
   - **Příklad:** Windows NT, Minix.
3. Hybridní jádro:
   - Kombinuje prvky monolitického jádra a mikro jádra. Jádro obsahuje pouze základní funkce, zatímco ostatní služby běží v oddělených modulech.
   - **Příklad:** Moderní verze Windows, macOS.
4. Exojádro:

- Minimalistická architektura, která ponechává správu zdrojů a služeb uživatelským aplikacím. Často používáno ve výzkumném prostředí.

15.5. Architektura Windows

Architektura Windows je navržena jako **hybridní jádro**, které kombinuje prvky mikro jádra a monolitického jádra.
Komponenty architektury Windows:

1. User Mode (Uživatelský režim):
   - **Aplikace:** Programy, které běží na OS a komunikují s OS prostřednictvím aplikačního programového rozhraní (API).
   - **Subsystemy prostředí:** Implementují rozhraní různých operačních systémů, jako jsou Win32, POSIX a OS/2.
   - **Service Control Manager:** Spravuje služby, které běží na pozadí (např. systémové služby).
2. Kernel Mode (Režim jádra):
   - **Executive**: Obsahuje hlavní systémové služby, jako je správa paměti, procesů, I/O a bezpečnosti.
   - **Kernel (jádro)**: Spravuje prioritizaci procesů, přepínání vláken, synchronizaci a přidělování času procesoru.
   - **HAL (Hardware Abstraction Layer)**: Poskytuje abstraktní rozhraní mezi hardwarem a softwarem, což umožňuje operačnímu systému běžet na různých platformách.
   - **Device Drivers**: Řídí komunikaci mezi OS a hardwarem.

15.6. Architektura Linux

Linux je navržen jako **monolitické jádro** a je vysoce modulární, což umožňuje načítat a odstraňovat moduly jádra za běhu systému.
Komponenty architektury Linux:

1. User Space (Uživatelský prostor):
   - **Shell (příkazový interpret)**: Umožňuje uživateli spouštět příkazy a programy, přístup k souborům a správu systému.
   - **Uživatelské aplikace:** Aplikace, které běží v uživatelském režimu, komunikují s jádrem prostřednictvím systémových volání.
2. Kernel Space (Jádrový prostor):
   - **Správa procesů:** Linux podporuje multitasking a je schopen efektivně plánovat procesy a vlákna.
   - **Správa paměti:** Používá virtuální paměť a stránkování pro efektivní přidělování paměti.
   - **Správa souborového systému:** Linux podporuje různé souborové systémy (např. ext4, NTFS) a umožňuje přístup k souborům na různých úložných zařízeních.
   - **Device Drivers:** Ovladače pro hardware, které umožňují komunikaci mezi jádrem a fyzickými zařízeními.
   - **Modulární systém jádra:** Linux jádro je modulární, což umožňuje dynamicky načítat a odstraňovat moduly bez restartování systému.
3. Loadable Kernel Modules (LKM):
   - Umožňují přidávat funkce jádra, jako jsou nové ovladače nebo rozšíření bez nutnosti rekompilace nebo restartu systému.

15.7. Závěr

Architektura operačních systémů definuje, jak jsou organizovány a spravovány základní služby OS. Windows používá hybridní architekturu s oddělením uživatelského a jádrového prostoru, zatímco Linux má monolitické jádro s podporou modulů, které poskytují flexibilitu a stabilitu. Rozdílné typy OS a architektur odpovídají různým potřebám a preferencím, a to jak pro desktopové, tak pro serverové a vestavěné systémy.

16. Správa paměti

Paměťový a adresný prostor, metody přidělování paměti, virtuální paměť, správa paměti v OS MS Windows a Linux

16.1. Paměťový a adresný prostor

Paměťový prostor představuje množinu všech paměťových adres, které může proces používat k ukládání a načítání dat. Každý proces v operačním systému má přidělen vlastní paměťový prostor, což umožňuje izolaci procesů a zajišťuje bezpečnost dat.
Adresný prostor je množina adres, které jsou dostupné pro jednotlivé procesy v rámci operačního systému. Adresný prostor může být:

- **Fyzický adresný prostor**: Představuje skutečné fyzické adresy v RAM, na které procesy zapisují nebo čtou data.
- **Virtuální adresný prostor**: Poskytuje každému procesu vlastní logický adresní prostor, který je mapován na fyzickou paměť. Umožňuje využití větší paměti, než kolik je fyzicky dostupné, díky správě virtuální paměti.

16.2. Metody přidělování paměti

Operační systém spravuje přidělování paměti pomocí různých metod, které zajišťují efektivní využití paměti a minimalizují fragmentaci.
A) Kontinuální přidělování paměti

- **Single-Partition Allocation (Jednoduché přidělování)**: Celá paměť je přidělena jednomu procesu, což je vhodné pro jednoúlohové systémy, ale ne pro víceúlohové OS.
- **Multiple-Partition Allocation (Vícepartiční přidělování)**: Paměť je rozdělena na více bloků (partition), které jsou přiděleny jednotlivým procesům. Zajišťuje lepší využití paměti než single-partition.

B) Rozdělování paměti do pevných bloků (Fixed Partitioning)

- **Princip**: Paměť je rozdělena do pevných bloků různé velikosti, které jsou přiděleny procesům.
- **Výhody**: Jednoduchá implementace.
- **Nevýhody**: Vede k **vnitřní fragmentaci** (nevyužité místo uvnitř bloku, pokud proces potřebuje menší paměť).

C) Rozdělování paměti do proměnlivých bloků (Dynamic Partitioning)

- **Princip**: Paměť je rozdělena dynamicky podle požadavků procesů.
- **Výhody**: Efektivnější využití paměti než pevné bloky.
- **Nevýhody**: Může vést k **vnější fragmentaci** (rozdrobení paměti na menší volné bloky, které nejsou využitelné).

D) Stránkování (Paging)

- **Princip**: Fyzická paměť je rozdělena do pevných bloků zvaných **stránky**. Každý proces má vlastní stránkovou tabulku, která mapuje virtuální stránky na fyzické rámce.
- **Výhody**: Eliminuje vnější fragmentaci a umožňuje jednoduché přidělování paměti.
- **Nevýhody**: Může vést k režii spojené s přístupem k stránkovým tabulkám.

E) Segmentace

- **Princip**: Paměť je rozdělena do segmentů různé délky, které odpovídají logickým celkům programu (např. datové segmenty, kódy).
- **Výhody**: Umožňuje logickou organizaci paměti, lépe odpovídá struktuře programu.
- **Nevýhody**: Může vést k vnější fragmentaci.

F) Kombinace stránkování a segmentace

- **Princip**: Segmentace se používá pro logické členění programu, zatímco stránkování pro rozdělení segmentů do stránek.
- **Výhody**: Využívá výhod obou metod a minimalizuje fragmentaci.
- **Nevýhody**: Vyšší složitost správy paměti.

16.3. Virtuální paměť

Virtuální paměť umožňuje operačnímu systému využívat diskový prostor jako rozšíření fyzické paměti. Umožňuje, aby programy měly k dispozici větší adresný prostor než fyzická paměť.
Princip fungování virtuální paměti

1. **Stránkování a stránkovací soubor**: Virtuální paměť se implementuje pomocí stránkování, kde OS ukládá části paměti, které nejsou aktivně používány, na disk do tzv. stránkovacího (swap) souboru.
2. **Správa stránek**: Když proces potřebuje data, která nejsou v RAM, OS provede **page fault** a načte potřebnou stránku z disku do RAM, což může způsobit odložení jiné stránky zpět na disk.
3. **Výhody**: Umožňuje efektivní využití paměti, izolaci procesů a ochranu paměti.
4. **Nevýhody**: Časté přepínání stránek mezi RAM a diskem může vést k **thrashingu** (nadměrné zatížení disku), což výrazně snižuje výkon.

16.4. Správa paměti v OS MS Windows

MS Windows využívá stránkovanou virtuální paměť a několik mechanismů pro správu paměti, aby zajistil bezpečnost a stabilitu.

A) Virtuální adresní prostor

- Každý proces ve Windows má přidělen vlastní 4GB (u 32bit systémů) nebo větší (u 64bit systémů) virtuální adresní prostor.
- Virtuální paměť je rozdělena na části pro aplikace a pro jádro (např. 2 GB pro aplikace a 2 GB pro jádro na 32bit systému).

B) Stránkování a stránkovací soubor

- Fyzická paměť je rozdělena do stránek (obvykle 4 KB), které se mapují na fyzické rámce. Nepoužívané stránky se ukládají na disk do stránkovacího souboru (pagefile.sys).
- Správa paměti probíhá na základě algoritmu LRU (Least Recently Used) pro optimalizaci přístupu.

C) Správa a ochrana paměti

- **Mechanismus ochrany paměti**: Windows používá zabezpečený virtuální adresní prostor, který zajišťuje, že procesy nemohou přistupovat k paměti jiných procesů.
- **Memory-mapped files**: Windows podporuje souborově mapovanou paměť, která umožňuje sdílení dat mezi procesy bez nutnosti explicitního kopírování.

D) Speciální mechanismy

- **Copy-on-Write (COW)**: Používá se pro efektivní správu paměti při kopírování dat. Stránka je sdílena více procesy a kopírována až při zápisu do ní.
- **Přidělování paměti na vyžádání (Demand Paging)**: Windows načítá stránky do paměti pouze v případě potřeby, čímž optimalizuje využití RAM.

16.5. Správa paměti v OS Linux

Linux také používá stránkovanou virtuální paměť a nabízí pokročilé mechanismy pro optimalizaci správy paměti a sdílení mezi procesy.

A) Virtuální adresní prostor

- Linux, podobně jako Windows, přiděluje každému procesu vlastní virtuální adresní prostor, což umožňuje izolaci procesů a ochranu dat.
- Virtuální adresní prostor je rozdělen na část pro uživatelský prostor a část pro jádro.

B) Stránkování a stránkovací soubor

- Linux používá stránkovací soubor (swap space), který se může nacházet jako vyhrazený diskový oddíl nebo soubor na disku. Když paměť nestačí, používá disk jako rozšíření RAM.
- Implementace stránkování podporuje stránky o velikosti 4 KB (u většiny systémů).

C) Správa a ochrana paměti

- **Ochrana paměti**: Linux používá zabezpečený adresní prostor a zajišťuje, že procesy nemají přístup k paměti jiných procesů bez povolení.
- **Memory-mapped files**: Linux podporuje paměťově mapované soubory, což umožňuje efektivní sdílení a manipulaci s velkými datovými soubory mezi procesy.

D) Speciální mechanismy

- **Copy-on-Write (COW)**: Používá se při kopírování dat při forkování procesů. Stránky se kopírují až při pokusu o zápis.
- **Demand Paging**: Podobně jako ve Windows načítá Linux stránky do paměti na vyžádání.

- **Swappiness**: Nastavitelný parametr, který určuje, jak agresivně Linux používá swap. Vyšší hodnota způsobí častější swapování, nižší hodnota více využívá RAM.

E) Algoritmy správy paměti

- Linux využívá algoritmus **Least Recently Used (LRU)** k určení, které stránky mají být odloženy do swapu.
- Používá také **Transparent Huge Pages (THP)** pro optimalizaci práce s velkými datovými bloky, které přinášejí efektivnější přístup k paměti.

16.6. Závěr

Správa paměti je klíčovou funkcí každého operačního systému. Windows i Linux využívají stránkovanou virtuální paměť, která umožňuje efektivní přidělování paměti a ochranu dat. Oba systémy používají pokročilé techniky, jako je Copy-on-Write, demand paging a specifické algoritmy pro optimalizaci správy paměti. Virtuální paměť a různé metody přidělování paměti umožňují efektivní využití dostupných zdrojů a zajišťují stabilitu a bezpečnost systémů.

17. Procesy

evidence, běh procesů, správa front procesů, synchronizace procesů, uváznutí procesů

17.1. Procesy a evidence procesů

Co je proces?

- **Proces** je instance běžícího programu, který může obsahovat několik vláken a vyžaduje systémové zdroje (paměť, čas procesoru) pro svůj běh. Procesy jsou klíčovým prvkem multitaskingu v operačních systémech, který umožňuje běh více programů najednou.

Evidence procesů

- **PCB (Process Control Block)**: Každý proces je evidován v tzv. blokovém řízení procesů (PCB), což je datová struktura, kterou OS používá ke sledování všech informací o procesu.

Informace uložené v PCB zahrnují:

  - **Identifikátor procesu (PID)**: Unikátní identifikátor procesu.
  - **Stav procesu**: Označuje aktuální stav procesu (běží, čeká, ukončen).
  - **Programový čítač (PC)**: Ukládá adresu příští instrukce, která má být provedena.
  - **Registry CPU**: Uchovávají stav registrů, když proces není aktivní.
  - **Informace o paměti**: Obsahuje základní a mezní adresy paměti přidělené procesu.
  - **Seznam otevřených souborů**: Uvádí všechny soubory, ke kterým má proces přístup.

17.2. Běh procesů a jejich stavy

Procesy mohou přecházet mezi několika různými stavy během svého životního cyklu:

1. **Nový (New)**: Proces je vytvořen, ale ještě není připraven ke spuštění.
2. **Připravený (Ready)**: Proces čeká na přiřazení CPU, aby mohl být proveden. Je připraven k běhu, ale čeká ve frontě.
3. **Běžící (Running)**: Proces je právě prováděn na CPU.
4. **Čekající (Waiting/Blocked)**: Proces čeká na splnění určité podmínky nebo události (např. dokončení I/O operace).
5. **Ukončený (Terminated)**: Proces byl dokončen nebo ukončen a všechny jeho zdroje jsou uvolněny.

Proces může přecházet mezi těmito stavy v závislosti na akcích operačního systému a systému plánování.

17.3. Správa front procesů

Procesy se často nacházejí ve frontách podle svého aktuálního stavu. Operační systém spravuje různé fronty, které obsahují procesy připravené k běhu, čekající nebo blokované.
Typy front procesů

- **Fronta připravených procesů (Ready Queue)**: Obsahuje všechny procesy, které jsou připraveny k provádění, ale čekají na přiřazení CPU.
- **Fronty čekajících procesů (Waiting/Blocked Queue)**: Obsahují procesy, které čekají na dokončení určité události (např. vstup/výstup).
- **Fronta ukončených procesů**: Obsahuje všechny procesy, které byly dokončeny nebo ukončeny chybou.

Plánování procesů

- **Plánovač (Scheduler)**: Rozhoduje o přidělování procesoru jednotlivým procesům na základě určitých algoritmů plánování.

Základní algoritmy plánování zahrnují:

  - **FCFS (First-Come, First-Served)**: Procesy jsou prováděny v pořadí, ve kterém přijdou do fronty.
  - **SJF (Shortest Job First)**: Proces s nejkratší dobou provádění je vybrán jako první.
  - **Round Robin (RR)**: Každý proces dostane časový úsek a po jeho vypršení je zařazen zpět do fronty.
  - **Prioritní plánování**: Procesy jsou vybírány podle priority. Proces s vyšší prioritou dostane přednost před procesy s nižší prioritou.

17.4. Synchronizace procesů

Synchronizace procesů je klíčová pro správnou koordinaci procesů, které spolupracují nebo sdílejí zdroje. Synchronizace je nutná, aby se zabránilo vzniku chyb při současném přístupu k sdíleným datům nebo zdrojům.
Kritická sekce

- **Kritická sekce** je část kódu, kde proces přistupuje k sdíleným zdrojům. Pro správnou synchronizaci je nutné zajistit, aby do kritické sekce mohl vstoupit pouze jeden proces najednou.

Základní techniky synchronizace

1. Semafory
    - **Princip**: Semafor je proměnná, která slouží k řízení přístupu k sdíleným prostředkům. Má dvě hlavní operace – wait() a signal().
    - Typy semaforů:
        - **Binární semafory**: Mají pouze dvě hodnoty (0 a 1), které jsou podobné zamykacím mechanismům.
        - **Počítací semafory**: Umožňují více než jednomu procesu přístup ke zdroji.
2. Mutexy
    - **Princip**: Mutex (zkratka pro mutual exclusion) je zámek, který umožňuje exkluzivní přístup ke zdroji pouze jednomu procesu v daném čase.
    - **Využití**: Používají se v situacích, kde je třeba omezit přístup k citlivým datům nebo zdrojům na jeden proces najednou.
3. Monitory
    - **Princip**: Monitor je synchronizační nástroj, který zajišťuje, že pouze jeden proces může provádět operace s kritickou sekcí. Umožňuje použití podmínek a čekání/signalizaci na stav.
4. Podmíněné proměnné (Condition Variables)
    - **Princip**: Jsou to proměnné, které umožňují procesům čekat na splnění určité podmínky.
    - **Použití**: Proces čeká, dokud podmínka není splněna, a po splnění je probuzen signálem.

17.5. Uváznutí procesů (Deadlock)

Uváznutí (deadlock) je situace, kdy dva nebo více procesů čeká na zdroj, který je držen jiným procesem ve skupině, což způsobí, že žádný z procesů nemůže pokračovat. To může vést k trvalému zablokování procesů.
Podmínky pro vznik uváznutí (podle Coffmanových podmínek)

1. **Vzájemné vyloučení (Mutual Exclusion)**: Každý zdroj je buď přiřazen jednomu procesu, nebo je volný.
2. **Podmíněné čekání (Hold and Wait)**: Proces držící jeden zdroj může čekat na další zdroje.
3. **Neodnímatelnost (No Preemption)**: Zdroje nelze odebrat z procesu násilím; proces je musí uvolnit sám.
4. **Cykl čekání (Circular Wait)**: Existuje cyklus procesů, kde každý proces čeká na zdroj držený dalším procesem v cyklu.

Metody řešení uváznutí

1. **Prevence uváznutí**: Zajišťuje, aby alespoň jedna ze čtyř podmínek nutných pro vznik uváznutí nebyla splněna.
    - **Vyloučení podmíněného čekání**: Proces musí požádat o všechny zdroje najednou.
    - **Zamezení cyklického čekání**: Uspořádání zdrojů v pořadí a vyžadování, aby procesy žádaly zdroje podle tohoto pořadí.
2. Detekce a zotavení:
    - **Detekce uváznutí**: Systém monitoruje cykly čekání mezi procesy a detekuje, kdy vzniká uváznutí.
    - **Zotavení**: Procesy nebo zdroje jsou uvolněny, obvykle násilím, aby se odstranilo uváznutí (např. zabití procesu nebo odebrání zdroje).
3. Vyhýbání se uváznutí (Deadlock Avoidance):
    - **Bankéřův algoritmus**: Dynamicky analyzuje požadavky procesů a rozhoduje, zda může požadavek vést k bezpečnému stavu (bez uváznutí). Tento přístup je náročný na výpočet a je použitelný v systémech s nízkým počtem procesů a zdrojů.

17.6. Závěr

Správa procesů je klíčová pro efektivní a bezpečný běh více úloh současně. OS používá různé techniky pro evidenci a synchronizaci procesů a pro správu jejich přístupu k systémovým zdrojům. Synchronizace pomocí semaforů, mutexů a monitorů zajišťuje, že procesy přistupují k prostředkům bez kolizí. Uváznutí procesů je nežádoucí situace, které se lze vyhnout různými metodami, včetně prevence a detekce.

18. Souborové systémy

adresářová struktura, soubory a systémy souborů, souborové systémy ve MS Windows a Linux

18.1. Adresářová struktura

Adresářová struktura je způsob, jakým jsou soubory a složky organizovány v souborovém systému. Cílem adresářové struktury je umožnit uživatelům snadno vyhledávat a přistupovat k souborům a složkám.
Typy adresářových struktur

1. Jednoduchá struktura (Flat Directory):
    - Všechny soubory jsou uloženy na jedné úrovni, bez podadresářů.
    - **Nevýhody**: Rychle se stává nepřehlednou, pokud je mnoho souborů.
2. Hierarchická struktura (Tree Directory):
    - Vytváří se hierarchický strom složek, který obsahuje kořenový adresář a jeho podadresáře.
    - **Výhody**: Umožňuje efektivní organizaci souborů podle kategorií a podporuje víceúrovňové struktury.
    - **Příklady**: Adresářová struktura používaná v MS Windows, Linuxu a dalších moderních OS.
3. Acyklický graf (Acyclic Graph Directory):
    - Umožňuje vytvářet odkazy mezi adresáři, což znamená, že soubor nebo složka může existovat na více místech.
    - **Výhody**: Umožňuje sdílení souborů bez jejich kopírování.
    - **Nevýhody**: Komplikace při mazání a správě souborů.
4. General Graph Directory:
    - Rozšiřuje acyklickou strukturu o možnost vytváření cyklů (v praxi však není často implementována kvůli riziku nejednoznačností).

18.2. Soubory a systémy souborů

Soubor

- **Soubor** je logická jednotka dat uložených na úložném médiu, obsahující data (text, obrázky, programy atd.). Každý soubor má svůj **název** a **příponu**, která určuje typ souboru.

Systém souborů (file system)

- **Systém souborů** je struktura, kterou OS používá k organizaci, ukládání a správě souborů na úložném zařízení.
- Systémy souborů definují, jak jsou data uložena, jaká metadata jsou uchovávána (např. velikost, čas vytvoření) a jaký způsob přístupu je možný.

18.3. Základní typy systémů souborů

1. FAT (File Allocation Table):
    - Používá jednoduchou tabulku pro ukládání informací o umístění souborů na disku.
    - **Verze**: FAT12, FAT16, FAT32, exFAT.
    - **Výhody**: Široká kompatibilita, běžně používaný na flash discích a externích zařízeních.
    - **Nevýhody**: Omezení velikosti souboru na 4 GB (u FAT32), méně efektivní pro velké disky.
2. NTFS (New Technology File System):
    - Moderní souborový systém vyvinutý společností Microsoft, využívaný ve Windows.
    - **Výhody**: Podpora velkých souborů, vyšší bezpečnost (oprávnění k souborům), šifrování, komprese, možnost obnovy po chybách.
    - **Nevýhody**: Omezená kompatibilita s jinými OS než Windows.
3. Ext (Extended File System):
    - Souborový systém používaný v Linuxu, existuje v několika verzích (ext2, ext3, ext4).
    - **Ext2**: První verze ext s jednoduchou strukturou.
    - **Ext3**: Přidává podporu žurnálování, které chrání před poškozením dat.
    - **Ext4**: Podporuje větší velikosti souborů, lepší výkon a sníženou fragmentaci.
    - **Výhody**: Stabilní a efektivní pro Linux, podpora žurnálování.
    - **Nevýhody**: Omezená kompatibilita s Windows.
4. ReiserFS:

- Alternativní souborový systém pro Linux s efektivními metodami komprese a rychlým přístupem k malým souborům.
- **Výhody**: Rychlé ukládání a načítání souborů, podpora žurnálování.
- **Nevýhody**: Menší podpora než ext systémy.

5. APFS (Apple File System):
   - Moderní souborový systém používaný v macOS a iOS.
   - **Výhody**: Optimalizován pro flash paměti, podpora šifrování a rychlého klonování.
   - **Nevýhody**: Omezená kompatibilita mimo Apple zařízení.

18.4. Souborové systémy ve Windows

Windows používá především souborové systémy **NTFS**, **FAT32** a **exFAT**.
NTFS (New Technology File System)

- Vlastnosti:
  - **Žurnálování**: Pomáhá obnovit data v případě systémového pádu nebo výpadku.
  - **Zabezpečení**: Umožňuje nastavení oprávnění a šifrování souborů pomocí EFS (Encrypting File System).
  - **Komprese a šifrování**: NTFS umožňuje šifrování na úrovni souborů a složek a podporuje kompresi.
  - **Velikost souboru a disku**: NTFS umožňuje velké soubory a disky (teoreticky až do 16 EB).
  - **Diskové kvóty**: Umožňuje omezit množství úložiště dostupného jednotlivým uživatelům.
- **Použití**: NTFS je primární souborový systém pro pevné disky v moderních verzích Windows.

FAT32 a exFAT

- **FAT32**: Tradiční systém s širokou kompatibilitou, ale omezením velikosti souborů na 4 GB.
- **exFAT**: Vylepšená verze FAT32 s podporou větších souborů a disků, využívá se na flash pamětech a externích discích.

Adresářová struktura ve Windows

- **Kořenový adresář (Root Directory)**: Označen jako „C:\“ (nebo jiné písmeno), kde začíná adresářová struktura.
- Standardní složky:
  - C:\Program Files – pro instalované aplikace.
  - C:\Users – pro uživatelské složky a data.
  - C:\Windows – systémové soubory a knihovny.

18.5. Souborové systémy v Linuxu

Linux běžně používá souborové systémy **ext2**, **ext3**, **ext4**, ale může také podporovat alternativní systémy jako **ReiserFS**, **XFS** a další.
Ext4 (Fourth Extended File System)

- Vlastnosti:
  - **Žurnálování**: Ext4 podporuje žurnálování, což zajišťuje vyšší spolehlivost a rychlost při obnově po chybě.
  - **Velikost souboru a disku**: Podporuje soubory až do velikosti 16 TB a svazky až 1 EB.
  - **Zlepšený výkon**: Díky efektivní alokaci bloků a snížené fragmentaci.
- **Použití**: Ext4 je hlavním souborovým systémem používaným v mnoha distribucích Linuxu, jelikož poskytuje stabilitu a rychlost.

XFS

- **Vlastnosti**: Vhodný pro velké soubory, poskytuje vysoký výkon při paralelních operacích.
- **Použití**: Často se využívá na serverech a systémech s vysokými nároky na výkon a práci s velkými daty.

Btrfs (B-tree file system)

- **Vlastnosti**: Moderní souborový systém s podporou žurnálování, snapshotů a klonování souborů.
- **Výhody**: Vhodný pro správu velkých objemů dat a složitější úložné systémy.

Adresářová struktura v Linuxu

- Hierarchická struktura začínající v kořenovém adresáři /:
  - /bin – základní binární programy.
  - /home – domovské adresáře uživatelů.
  - /etc – konfigurační soubory systému.

- /var – proměnlivá data (logy, fronty).
- /lib – sdílené knihovny.
- /dev – soubory pro zařízení.
- /mnt a /media – pro připojené svazky (např. USB disky).

18.6. Porovnání souborových systémů Windows a Linux

Funkce	Windows (NTFS)	Linux (ext4)
Žurnálování	Ano	Ano
Šifrování	Ano (EFS)	Částečně (LUKS pro celý disk)
Velikost souboru	Teoreticky až 16 EB	Až 16 TB
Fragmentace	Může se vyskytovat	Minimalizována díky dynamické alokaci
Kompatibilita	Windows	Linux (s omezenou podporou ve Windows)
Práva souborů	Podpora oprávnění a auditování	Rozsáhlá podpora práv a oprávnění (chmod, chown)

18.7. Závěr

Souborové systémy a adresářové struktury jsou klíčovou součástí správy dat v operačních systémech. Windows a Linux používají odlišné souborové systémy (NTFS pro Windows a ext4 pro Linux) s různými funkcemi a možnostmi. Adresářová struktura v Linuxu je přísně hierarchická, zatímco Windows používá jednodušší strukturu s kořenem na oddílech označených písmeny. Oba OS poskytují robustní nástroje pro správu souborů, jejich oprávnění a bezpečnost.

19. Správa zařízení a sítě

Ovladače, paměťová média, operační paměť, správa sítě

19.1. Ovladače (Drivers)

Ovladače jsou softwarové komponenty, které umožňují operačnímu systému komunikovat s hardwarem. Každé zařízení, jako jsou tiskárny, grafické karty, síťové karty nebo úložná média, potřebuje ovladač, který poskytuje operačnímu systému rozhraní k danému zařízení.

Funkce ovladačů

- **Abstrakce hardwaru**: Ovladače skryjí detaily komunikace s hardwarem a poskytují standardizované rozhraní pro OS.
- **Přístup k zařízením**: Ovladače umožňují operačnímu systému číst data z zařízení a zapisovat data na zařízení.
- **Správa zařízení**: Řídí provoz zařízení (např. řízení rychlosti tiskárny) a zajišťují, že se hardware chová očekávaným způsobem.

Typy ovladačů

1. **Jádrové (kernel) ovladače**: Tyto ovladače pracují na úrovni jádra operačního systému a mají přímý přístup k hardwaru.
2. **Uživatelské ovladače (user-mode drivers)**: Běží v uživatelském prostoru a využívají jádro jako prostředníka pro komunikaci s hardwarem.

Příklady ovladačů

- **Ovladače pro grafické karty (GPU)**: Zajišťují, že OS může komunikovat s grafickými kartami a podporují zobrazování grafiky.
- **Síťové ovladače (NIC drivers)**: Umožňují komunikaci mezi OS a síťovými kartami, čímž podporují připojení k síti.
- **Ovladače pro úložná média (např. SATA, NVMe)**: Řídí přístup k pevným diskům, SSD a dalším paměťovým zařízením.

Správa ovladačů

- **Instalace a aktualizace**: OS poskytují nástroje pro instalaci a aktualizaci ovladačů. Aktualizace často řeší bezpečnostní chyby nebo zvyšují výkon zařízení.
- **Kompatibilita**: Ovladače by měly být kompatibilní s verzí OS a hardwarem.
- **Diagnostika a oprava**: Moderní OS nabízejí nástroje pro diagnostiku problémů s ovladači a možnost odinstalace nebo přeinstalace nefunkčních ovladačů.

19.2. Paměťová média

Paměťová média jsou zařízení, která umožňují ukládání a načítání dat. Tato média se liší podle kapacity, rychlosti, typu ukládání dat a způsobu připojení.

Typy paměťových médií

1. Pevný disk (HDD):
    - Mechanické úložné zařízení využívající magnetické plotny a čtecí/zapisovací hlavy.
    - **Výhody**: Vysoká kapacita, nižší cena za GB.
    - **Nevýhody**: Pomalejší než SSD, náchylný k mechanickému poškození.
2. Solid State Drive (SSD):
    - Flash úložiště, které nemá pohyblivé části.
    - **Výhody**: Rychlé načítání dat, nízká latence, odolnost proti mechanickému poškození.
    - **Nevýhody**: Vyšší cena za GB než HDD.
3. USB flash disk:
    - Přenosné médium s flash pamětí, připojitelné přes USB.
    - **Výhody**: Malá velikost, přenositelnost.
    - **Nevýhody**: Omezená kapacita ve srovnání s HDD a SSD, omezená životnost.
4. Paměťová karta (SD, microSD):
    - Používaná v mobilních zařízeních, kamerách, tabletech.
    - **Výhody**: Malá velikost, odolnost.
    - **Nevýhody**: Omezená kapacita, nižší rychlost než SSD.

5. Optická média (CD, DVD, Blu-ray):
    - Používaná pro uchování dat na disku čitelném laserem.
    - **Výhody**: Dlouhodobá životnost, odolnost vůči elektromagnetickému rušení.
    - **Nevýhody**: Omezená kapacita, vyžaduje optickou mechaniku.

Správa paměťových médií

- **Formátování**: Proces přípravy média pro použití s určitou strukturou souborového systému (např. NTFS, exFAT).
- **Optimalizace**: Např. defragmentace HDD nebo TRIM u SSD, které zajišťují optimální výkon média.
- **Zálohování a obnova**: OS nabízí nástroje pro zálohování a obnovu dat na paměťových médiích.

19.3. Operační paměť

Operační paměť (RAM) je klíčová složka počítače, která poskytuje dočasné úložiště pro spuštěné programy a systémové procesy. RAM umožňuje rychlý přístup k datům, ale je volatilní, což znamená, že data se po vypnutí ztratí.
Typy operační paměti

1. **DRAM (Dynamic RAM)**: Vyžaduje pravidelný refresh dat.
    - Používá se ve většině počítačů jako hlavní paměť.
2. **SRAM (Static RAM)**: Data zůstávají zachována bez potřeby obnovování, je rychlejší než DRAM.
    - Využívá se v cache paměti procesorů.
3. **DDR (Double Data Rate)**: Typ DRAM, který umožňuje přenos dat dvakrát za taktovací cyklus.
    - Moderní verze zahrnují DDR4 a DDR5.

Správa operační paměti

1. **Přidělování paměti**: OS přiděluje paměť procesům na základě jejich potřeby. Využívá mechanismy jako **virtuální paměť**, která umožňuje využít disk jako rozšíření RAM.
2. **Stránkování a segmentace**: Používají se k rozdělení paměti na menší části pro efektivní správu a ochranu paměti mezi procesy.
3. **Optimalizace pomocí cache**: OS využívá cache paměť (SRAM) pro uložení často používaných dat, což zvyšuje výkon.
4. **Swapping**: Přesouvání neaktivních částí paměti na disk do stránkovacího souboru nebo swap oddílu.

19.4. Správa sítě

Správa sítě zahrnuje všechny činnosti a nástroje, které OS používá k řízení, monitorování a údržbě síťových připojení a komunikace.
Síťové protokoly
Protokoly umožňují komunikaci mezi zařízeními a zajišťují strukturu datového přenosu.

1. **TCP/IP (Transmission Control Protocol/Internet Protocol)**: Základní sada protokolů pro komunikaci v sítích a na internetu.
    - **TCP** zajišťuje spolehlivý přenos dat mezi zařízeními.
    - **IP** zajišťuje adresování a směrování paketů.
2. **UDP (User Datagram Protocol)**: Rychlý, ale nespolehlivý protokol pro datové přenosy, které nevyžadují potvrzení (např. streamování).
3. **DNS (Domain Name System)**: Překlad doménových jmen na IP adresy.
4. **DHCP (Dynamic Host Configuration Protocol)**: Automaticky přiděluje IP adresy zařízení v síti.

Síťová zařízení a správa

- **Router**: Zajišťuje komunikaci mezi sítěmi, směrování paketů mezi IP adresami.
- **Switch**: Propojuje zařízení v rámci stejné sítě a umožňuje jim komunikovat.
- **Firewall**: Blokuje nebo povoluje síťový provoz na základě bezpečnostních pravidel, chrání síť před neoprávněnými přístupy.

Správa síťových připojení v OS

1. **Konfigurace IP adresy**: OS umožňuje manuální nastavení IP adresy, masky podsítě a brány nebo použití DHCP pro automatickou konfiguraci.
2. **Správa síťových služeb**: OS poskytuje nástroje pro správu síťových služeb, jako jsou DNS, FTP, HTTP, SSH.
3. **Monitorování sítě**: Nástroje jako **ping**, **traceroute**, **netstat** a **Wireshark** slouží k diagnostice problémů v síti.
4. Bezpečnostní opatření:

- **Firewally**: Zabraňují neoprávněnému přístupu k síťovým prostředkům.
- **Šifrování**: Šifrování datových přenosů pomocí protokolů (např. TLS) zajišťuje bezpečnost dat.

19.5. Závěr

Správa zařízení a sítě zahrnuje efektivní řízení hardwarových prostředků, jako jsou paměťová média a RAM, spolu s robustní správou síťových připojení a komunikace. Ovladače hrají klíčovou roli v propojení OS a hardwaru, zatímco síťová správa zajišťuje spolehlivé a bezpečné připojení mezi zařízeními. Moderní operační systémy využívají pokročilé techniky a protokoly pro optimalizaci výkonu a zajištění bezpečnosti v sítích i při správě zařízení.

20. Správa periferií

I/O systém, druhy periferií, přerušení, ovladače, bloková zařízení

20.1. I/O systém (Vstupně-výstupní systém)

I/O systém (Input/Output system) v operačním systému zajišťuje komunikaci mezi procesorem, pamětí a periferními zařízeními. Systém zajišťuje, že data mohou proudit mezi periferiemi a hlavní pamětí (RAM) nebo procesorem bez zásahů uživatele.

Funkce I/O systému

- **Komunikace mezi OS a hardwarem**: I/O systém poskytuje rozhraní pro komunikaci mezi softwarem a periferními zařízeními.
- **Řízení periferií**: Zajišťuje, že zařízení dostanou požadované příkazy, a monitoruje jejich stav.
- **Přístup ke sdíleným zdrojům**: Řídí přístup více procesů k periferiím, aby nedocházelo k zablokování nebo konfliktům.
- **Přenos dat**: Provádí efektivní přesun dat mezi zařízením a pamětí (např. pomocí DMA - Direct Memory Access).

Metody přenosu dat

1. **Programově řízený vstup/výstup**: CPU přímo ovládá zařízení, což je jednoduché, ale neefektivní, protože CPU musí čekat na dokončení každé operace.
2. **Přímý přístup do paměti (DMA)**: Umožňuje přenos dat mezi periferiemi a pamětí bez přímé účasti CPU, což šetří čas procesoru.
3. **Přerušení**: Zařízení signalizuje procesoru potřebu pozornosti prostřednictvím přerušení, což umožňuje efektivní sdílení zdrojů.

20.2. Druhy periferií

Periferie lze rozdělit podle jejich účelu a způsobu připojení k systému.

Klasifikace periferií

1. Vstupní zařízení:
    - Umožňují uživateli zadávat data do systému.
    - **Příklady**: Klávesnice, myš, skener, mikrofon.
2. Výstupní zařízení:
    - Umožňují systému komunikovat data uživateli.
    - **Příklady**: Monitor, tiskárna, reproduktory.
3. Vstupně-výstupní zařízení (obousměrná):
    - Umožňují obousměrnou komunikaci dat.
    - **Příklady**: Diskové jednotky, flash disky, síťové karty.

Podle připojení k systému

1. **Připojení přes kabel** (USB, SATA, HDMI): Fyzicky připojené zařízení, která zajišťují stabilní a rychlý přenos dat.
2. **Bezdrátové připojení** (Wi-Fi, Bluetooth): Zařízení komunikují s počítačem bezdrátově, což je výhodné pro mobilitu, ale může být náchylnější k rušení.

Další rozdělení periferií podle funkcí

1. **Bloková zařízení**: Ukládají data v blocích a umožňují náhodný přístup k datům.
    - **Příklady**: Pevné disky, SSD, optické jednotky.
2. **Znaková zařízení**: Přenášejí data po znacích (byt po bytu) a mají sekvenční přístup.
    - **Příklady**: Klávesnice, sériové porty, terminály.

20.3. Přerušení

Přerušení (interrupt) je signál od periferního zařízení k procesoru, který informuje o události vyžadující pozornost. Přerušení umožňují procesoru přerušit aktuální činnost, aby se mohl věnovat obsluze periferie, a poté se vrátit k původnímu úkolu.

Typy přerušení

1. Hardwarová přerušení:
    - Generována externími zařízeními (např. klávesnicí nebo síťovou kartou).
    - Procesor okamžitě přeruší svou činnost a spustí obslužnou rutinu přerušení.

2. Softwarová přerušení:
    - Generována softwarem, když program potřebuje přístup k systémovým službám.
    - Používá se pro interní operace, například přístup k operačnímu systému.
3. Maskovatelné a nemaskovatelné přerušení:
    - **Maskovatelné**: Přerušení, která lze zakázat, pokud je to nezbytné.
    - **Nemaskovatelné**: Kritická přerušení, která nelze zakázat, protože jsou nezbytná pro stabilitu systému (např. chyba paměti).

Princip fungování přerušení

1. **Generování přerušení**: Zařízení odešle přerušovací signál, který zastaví aktuální instrukci procesoru.
2. **Zpracování přerušení**: Procesor uloží stav běžícího procesu, aby se k němu mohl později vrátit, a přejde k obslužné rutině přerušení.
3. **Obsluha přerušení**: Specifická část OS (interrupt handler) řeší přerušení a provádí nezbytné kroky (např. zpracování dat ze zařízení).
4. **Návrat k původnímu úkolu**: Po vyřešení přerušení se procesor vrátí k původní instrukci a pokračuje v její vykonávání.

20.4. Ovladače

Ovladače periferií jsou softwarové moduly, které umožňují operačnímu systému komunikovat s hardwarem. Ovladače mapují specifické požadavky zařízení do formátu, který OS dokáže zpracovat.
Typy ovladačů

1. Ovladače na úrovni jádra (Kernel-mode drivers):
    - Mají přímý přístup k hardwaru a jádru OS, což jim umožňuje rychlý přístup k prostředkům.
    - Používají se pro základní hardwarová zařízení, jako jsou síťové a grafické karty.
2. Uživatelské ovladače (User-mode drivers):
    - Běží mimo jádro OS, což snižuje riziko selhání systému při chybě ovladače.
    - Používají se pro méně kritické periferie, jako jsou tiskárny.

Funkce ovladačů

- **Přístup k zařízením**: Poskytují OS a aplikacím přístup k periferním zařízením.
- **Zpracování přerušení**: Ovladače spolupracují s přerušením, aby mohly obsloužit zařízení, když je potřeba.
- **Přidělování paměti**: Ovladače často spravují paměť zařízení a alokují prostředky potřebné pro komunikaci.

20.5. Bloková zařízení

Bloková zařízení jsou periferní zařízení, která čtou nebo zapisují data ve formě bloků pevné velikosti. Tato zařízení umožňují náhodný přístup k jednotlivým blokům dat, což znamená, že data mohou být čtena nebo zapisována z libovolného místa na zařízení.
Příklady blokových zařízení

- **Pevné disky (HDD)**: Data jsou rozdělena do sektorů a lze k nim přistupovat v libovolném pořadí.
- **SSD (Solid State Drive)**: Rychlejší než HDD a podporují náhodný přístup k datům.
- **Optické disky (CD, DVD)**: Data jsou uložena na kruhových stopách a mohou být přístupná v blocích.

Vlastnosti blokových zařízení

- **Náhodný přístup**: Umožňuje přímý přístup k jakémukoli bloku dat na zařízení, což je efektivní pro rychlý přístup k velkým souborům.
- **Vysoká kapacita**: Bloková zařízení obvykle mají vysokou kapacitu pro ukládání dat, což je výhodné pro dlouhodobé ukládání velkého množství dat.
- **Vhodné pro žurnálování**: Podporují souborové systémy s žurnálováním, což zvyšuje spolehlivost při práci s velkým množstvím dat.

20.6. Závěr

Správa periferií je klíčová pro efektivní využití hardwarových zdrojů v počítačovém systému. Operační systém používá I/O systém a ovladače k zajištění komunikace s periferiemi. Přerušení umožňují efektivní sdílení zdrojů mezi CPU a periferiemi. Bloková zařízení poskytují vysokokapacitní úložiště s možností náhodného přístupu, což je nezbytné pro moderní úložné systémy. Tímto způsobem OS zajišťuje efektivní a bezpečné využití všech dostupných periferií a zdrojů.

21. Řízení přístupu a práva uživatelů

základní pojmy související s oprávněním, uživatelské profily, účty a skupiny, přístupová oprávnění, objekty, zásady, skupiny

21.1. Základní pojmy související s oprávněním

Řízení přístupu je proces, který zajišťuje, že pouze oprávněné osoby mají přístup k určitému prostředku nebo datům. Řízení přístupu se zakládá na určení a kontrole, kdo může vidět, měnit nebo provádět akce na konkrétních objektech v rámci systému.

- **Autentizace**: Proces ověření identity uživatele, obvykle prostřednictvím hesla, biometrických údajů nebo multifaktorového ověření (např. kombinace hesla a SMS).
- **Autorizace**: Proces přidělení přístupových práv uživateli na základě jeho role nebo potřeb. Autorizace určuje, ke kterým zdrojům a operacím má uživatel přístup.
- **Oprávnění (Permissions)**: Určují, co může uživatel s daným objektem provádět (např. číst, zapisovat, mazat).
- **Princip minimálních oprávnění**: Zásada, že uživatelé a procesy by měli mít jen ta oprávnění, která jsou nezbytná k plnění jejich úkolů.
- **ACL (Access Control List)**: Seznam pravidel přístupu, která určují, jaká oprávnění má každý uživatel nebo skupina k určitému objektu.

21.2. 2. Uživatelské profily, účty a skupiny

Uživatelský profil je sada informací a nastavení, která definují prostředí uživatele v operačním systému. Profil obvykle zahrnuje data, jako jsou soubory na ploše, nastavení prohlížeče a konfigurace aplikací.

Uživatelské účty

- **Uživatelský účet** je jedinečná identita, pod kterou se uživatel přihlašuje do systému. Účet umožňuje rozlišení mezi různými uživateli, přiřazení oprávnění a monitorování aktivit.
    - **Standardní účet**: Uživatel s omezeným přístupem k systémovým zdrojům, který může provádět běžné úkoly.
    - **Administrátorský účet**: Má plná oprávnění k systému, což zahrnuje instalaci softwaru, změnu systémových nastavení a správu jiných uživatelských účtů.

Skupiny uživatelů

- **Skupina** je kolekce uživatelských účtů, které mají společná oprávnění. Skupiny se využívají k zjednodušení správy oprávnění, protože umožňují přiřadit oprávnění celé skupině najednou, namísto jednotlivých uživatelů.
    - Příklady skupin:
        - **Administrátoři**: Mají přístup ke všem zdrojům a funkcím systému.
        - **Hosté**: Mají omezený přístup, obvykle jen k základním funkcím.
        - **Uživatelé s omezeným přístupem**: Mají přístup pouze k určitým aplikacím nebo složkám.

Uživatelský profil

- **Profil uživatele** uchovává data a nastavení specifická pro daného uživatele. Obsahuje například:
    - Konfiguraci prostředí a zobrazení pracovní plochy.
    - Přístupová práva ke sdíleným souborům a složkám.
    - Nastavení aplikací, například webového prohlížeče nebo kancelářských programů.

21.3. 3. Přístupová oprávnění

Přístupová oprávnění jsou specifická oprávnění přiřazená uživatelským účtům nebo skupinám a určují, jaký typ přístupu mají k objektům v systému.

Typy přístupových oprávnění

1. **Čtení (Read)**: Uživatel může zobrazit obsah souboru nebo složky, ale nemůže ho měnit.
2. **Zápis (Write)**: Uživatel může měnit obsah souboru nebo složky.
3. **Spouštění (Execute)**: Uživatel může spouštět programy nebo skripty.
4. **Úplný přístup (Full Control)**: Uživatel má veškerá práva k objektu, včetně jeho mazání nebo změny oprávnění pro ostatní.

Dědičnost oprávnění

- Složky mohou mít nastavená oprávnění, která se automaticky aplikují na jejich podřízené objekty, jako jsou soubory nebo podsložky. Tento mechanismus se nazývá **dědičnost**.

ACL (Access Control List)

- Každý objekt může mít připojenou seznam pravidel přístupu (ACL), kde jsou uvedena oprávnění pro jednotlivé uživatele nebo skupiny. ACL určuje, jaká oprávnění mají různé subjekty k danému objektu.

21.4. Objekty

V kontextu řízení přístupu je **objekt** jakýkoli zdroj, ke kterému mohou být přiřazena přístupová oprávnění.
Typy objektů

1. **Soubory a složky**: Základní objekty, ke kterým je běžně přístup řízen pomocí oprávnění.
2. **Síťové zdroje**: Sdílené složky, tiskárny nebo služby dostupné přes síť.
3. **Aplikace a procesy**: Přístup k aplikacím může být řízen oprávněními, což může zahrnovat například omezení přístupu k citlivým datům v aplikacích.
4. **Registry (v systému Windows)**: Systémový úložný prostor pro konfigurace, kde mohou být nastavena oprávnění k jednotlivým klíčům.

Práva a oprávnění k objektům

- Oprávnění k objektům určuje, jaký typ přístupu mají uživatelé k danému objektu. Tato oprávnění mohou být nastavena na úrovni jednotlivých uživatelů nebo skupin.

21.5. Zásady

Zásady (Policies) jsou pravidla a konfigurace, které definují, jak se systém chová a jaká jsou přístupová práva a oprávnění pro uživatele.
Typy zásad

1. **Zásady zabezpečení**: Řídí bezpečnostní aspekty, jako jsou minimální délka hesla, zamykání účtů po neúspěšných přihlášeních a expirace hesel.
2. **Zásady řízení přístupu**: Určují, které skupiny nebo uživatelé mají přístup ke konkrétním zdrojům. Mohou zahrnovat ACL, dědičnost a explicitní oprávnění.
3. **Zásady skupin (Group Policies)**: Používané především v doménových sítích (např. v prostředí Active Directory) k centrálnímu řízení konfigurace a oprávnění pro uživatele a počítače.
    - **Příklad**: Nastavení povinného složitého hesla, zamezení přístupu k určitému softwaru, instalace aktualizací.

Úrovně zásad

- **Lokální zásady**: Ovlivňují pouze jeden konkrétní počítač nebo uživatele.
- **Doménové zásady**: Platí pro všechny počítače a uživatele v síťové doméně.

21.6. Skupiny

Skupiny umožňují snadnou správu oprávnění tím, že seskupují uživatele se stejnými potřebami přístupu k určitým zdrojům nebo objektům. Skupiny zjednodušují správu, protože správce může přidělit oprávnění celé skupině namísto jednotlivých uživatelů.
Typy skupin

1. Skupiny zabezpečení (Security Groups):
    - Používají se k řízení přístupu k objektům a definují, kdo má jaká oprávnění k souborům, aplikacím a jiným zdrojům.
2. Distribuční skupiny (Distribution Groups):
    - Slouží ke správě seznamů pro e-mailovou komunikaci, ale obvykle nemají žádný vliv na přístupová oprávnění.
3. Lokální a globální skupiny:
    - **Lokální skupiny**: Působí jen na jednom konkrétním počítači nebo zařízení.
    - **Globální skupiny**: Mohou být použity v celém prostředí domény.

Přiřazování práv pomocí skupin

- Skupiny umožňují, aby uživatelé měli specifická oprávnění a přístupová práva na základě členství ve skupině.
- Při přidání uživatele do skupiny zdědí všechna oprávnění této skupiny.

21.7. Závěr

Řízení přístupu a správa uživatelských oprávnění jsou základní prvky zabezpečení v operačních systémech a síťových prostředích. Uživatelské účty a skupiny umožňují organizovat uživatele podle jejich přístupových potřeb. ACL a zásady zabezpečení dále řídí, co mohou jednotliví uživatelé dělat s konkrétními objekty, což zajišťuje, že k citlivým zdrojům mají přístup pouze oprávněné osoby. Důležitou rolí v těchto systémech hraje princip minimálních oprávnění, který zvyšuje celkovou bezpečnost systému.

22. Nasazení systému MS Windows

instalace, aktualizace, registr Windows, chyby při běhu, start systému, správa systému, licencování

22.1. Instalace systému MS Windows

Předinstalační příprava

- **Požadavky na hardware**: Ujistěte se, že počítač splňuje minimální požadavky na CPU, RAM, diskový prostor a grafiku.
- **Záloha dat**: Doporučuje se zálohovat data, protože instalace může vyžadovat formátování disku.
- **Výběr edice Windows**: Vyberte správnou edici systému (Home, Pro, Enterprise) na základě potřeb uživatele nebo organizace.

Proces instalace

1. **Příprava instalačního média**: Instalační médium lze připravit pomocí USB disku, DVD nebo síťového instalačního serveru.
2. **Bootování z média**: Po spuštění počítače z instalačního média je nutné vybrat jazyk a další regionální nastavení.
3. Výběr typu instalace:
    - **Čistá instalace**: Všechny soubory na disku budou vymazány a systém bude nainstalován na formátovaný disk.
    - **Aktualizace (upgrade)**: Předchozí verze systému Windows bude aktualizována, přičemž uživatelská data a aplikace zůstanou zachovány.
4. Konfigurace instalace:
    - Výběr disku a nastavení oddílů.
    - Nastavení uživatelského účtu a případná volba nastavení účtu Microsoft.
    - Nastavení zabezpečení, síťového připojení a regionálních nastavení.

22.2. Aktualizace systému MS Windows

Aktualizace systému Windows jsou důležité pro udržení bezpečnosti, výkonu a stability systému. Aktualizace zahrnují bezpečnostní záplaty, opravy chyb a vylepšení funkcí.
Typy aktualizací

1. **Bezpečnostní aktualizace**: Zajišťují ochranu proti novým bezpečnostním hrozbám.
2. **Kumulativní aktualizace**: Obsahují všechny předchozí aktualizace, což umožňuje jejich rychlejší instalaci.
3. **Funkční aktualizace (Feature Updates)**: Představují novou verzi systému Windows s novými funkcemi (např. aktualizace na verzi 21H2).
4. **Volitelné aktualizace**: Doporučené, ale nejsou nutné (např. aktualizace ovladačů).

Způsoby aktualizace

- **Windows Update**: Základní nástroj pro získávání a instalaci aktualizací.
- **WSUS (Windows Server Update Services)**: Nástroj pro správu aktualizací ve firemním prostředí, který umožňuje řídit distribuci aktualizací.
- **Manual Updates**: Stahování a instalace aktualizací přímo ze stránek společnosti Microsoft.

Řešení problémů s aktualizacemi

- **Windows Update Troubleshooter**: Nástroj pro řešení běžných problémů s aktualizacemi.
- **Resetování Windows Update komponent**: Pomocí příkazového řádku lze obnovit komponenty aktualizací.
- **Odstranění aktualizace**: Pokud aktualizace způsobí problém, lze ji odstranit z ovládacího panelu.

22.3. Registr Windows

Registr Windows je hierarchická databáze, která uchovává nastavení a konfigurace systému, aplikací a uživatelských profilů.
Struktura registru

- **HKEY_CLASSES_ROOT (HKCR)**: Obsahuje informace o asociacích souborů a registraci objektů COM.
- **HKEY_CURRENT_USER (HKCU)**: Uchovává nastavení pro aktuálně přihlášeného uživatele.
- **HKEY_LOCAL_MACHINE (HKLM)**: Zahrnuje konfigurace hardwaru a software celého systému.

- **HKEY_USERS (HKU)**: Obsahuje data všech uživatelů, kteří mají účet na daném zařízení.
- **HKEY_CURRENT_CONFIG (HKCC)**: Obsahuje informace o aktuální konfiguraci hardwaru.

Práce s registrem

- **Regedit**: Hlavní nástroj pro prohlížení a úpravu registru. Provádění změn v registru vyžaduje opatrnost, protože nesprávná změna může způsobit nestabilitu systému.
- **Export a import**: Klíče a podklíče v registru lze exportovat (zálohovat) a následně importovat.

Zálohování a obnova registru

- **Bod obnovení systému**: Umožňuje obnovit stav systému, včetně registru, do předchozího bodu v čase.
- **Export registru**: Vytváří záložní kopii jednotlivých klíčů před provedením změn.

22.4. Chyby při běhu

Chyby při běhu systému Windows mohou být způsobeny chybami v aplikacích, konflikty s ovladači nebo problémy s hardwarem.

Typy chyb

1. **Modrá obrazovka smrti (BSOD)**: Kritická chyba systému, obvykle způsobená hardwarovými problémy nebo nekompatibilními ovladači.
2. **Skriptové chyby**: Obvykle se objevují při práci s webovými aplikacemi nebo v prohlížečích.
3. **Runtime Error**: Chyby způsobené konflikty v aplikacích nebo nesprávnými nastaveními.

Diagnostika chyb

- **Prohlížeč událostí (Event Viewer)**: Umožňuje procházet a analyzovat záznamy o chybách a událostech.
- **Správce zařízení (Device Manager)**: Diagnostikuje problémy s ovladači a zařízeními.
- **Safe Mode (Nouzový režim)**: Umožňuje spuštění systému s minimálním nastavením a diagnostiku problémů, které vznikají při běžném spuštění.

22.5. Start systému

Proces startu systému zahrnuje několik kroků, které vedou k plnému spuštění operačního systému.

Fáze startu systému Windows

1. **Inicializace hardwaru (BIOS/UEFI)**: Základní deska spustí BIOS nebo UEFI, který provede kontrolu hardwaru (POST) a spustí zavaděč systému.
2. **Zavaděč systému (Bootloader)**: Windows využívá Boot Manager (bootmgr), který načítá OS.
3. **Kernel Initialization**: Windows kernel se načte do paměti, inicializují se ovladače a systémové procesy.
4. **Session Initialization**: Windows inicializuje uživatelské přihlášení a další služby.
5. **User Logon**: Po přihlášení uživatele se spouští nastavení pracovního prostředí uživatele.

Řešení problémů se startem

- **Startup Repair**: Automatický nástroj pro opravu problémů se spuštěním systému.
- **Bootrec**: Příkazový nástroj, který umožňuje opravit MBR (Master Boot Record), BCD (Boot Configuration Data) a další problémy se startem.
- **Safe Mode**: Nouzový režim pomáhá odstranit problémy spuštěním systému s minimálním počtem ovladačů a služeb.

22.6. Správa systému

Správa systému zahrnuje různé úlohy, jako je správa uživatelů, konfigurace sítě, zabezpečení systému a zálohování.

Nástroje pro správu systému

- **Správce úloh (Task Manager)**: Poskytuje přehled o spuštěných procesech, výkonu systému a umožňuje ukončení aplikací.
- **Správce zařízení (Device Manager)**: Umožňuje kontrolu a konfiguraci hardwarových zařízení, aktualizaci ovladačů a řešení konfliktů hardwaru.
- **Místní zásady zabezpečení (Local Security Policy)**: Nástroj pro nastavení bezpečnostních pravidel, jako jsou zásady hesel nebo řízení přístupu.
- **Ovládací panely a Nastavení**: Centrální přístup k nastavení systému, který umožňuje přizpůsobit vzhled, síť, uživatele, bezpečnost atd.

Zálohování a obnova

- **Bod obnovení systému**: Umožňuje vrátit systém do dřívějšího stavu bez ztráty dat.
- **Obnova systému (System Restore)**: Obnovuje stav systému na základě bodu obnovení.
- **BitLocker**: Nástroj pro šifrování disku, který chrání data před neoprávněným přístupem.

22.7. Licencování systému Windows

Licencování systému Windows je klíčovým prvkem zajišťujícím legální používání softwaru a zamezení pirátství.
Typy licencí

1. OEM (Original Equipment Manufacturer):
    - Je určena pro jedno zařízení a je pevně spjata s hardwarem.
    - Licence je neodstranitelná a nelze ji přenést na jiný počítač.
2. Retail (krabicová verze):
    - Umožňuje přenos na jiné zařízení, pokud je systém odstraněn z původního.
    - Zahrnuje podporu od společnosti Microsoft.
3. Volume Licensing:
    - Určena pro organizace, které potřebují spravovat větší počet licencí.
    - Umožňuje centrální správu licencí a používání na několika zařízeních v rámci jedné smlouvy.

Aktivace systému Windows

- **Product Key**: Každá licence je ověřena pomocí unikátního klíče.
- **Digitální licence**: Licence je vázaná na účet Microsoft nebo na hardware a není nutné zadávat klíč při každé nové instalaci.
- **KMS (Key Management Service)** a **MAK (Multiple Activation Key)**: Používané v organizacích, které potřebují aktivovat více zařízení.

Výhody legální licence

- Pravidelné aktualizace a bezpečnostní záplaty.
- Záruka a podpora od společnosti Microsoft.
- Přístup ke všem funkcím systému a ochrana před omezeními, která mohou být aplikována na nelegální verze systému.

22.8. Závěr

Nasazení systému Windows zahrnuje řadu kroků, od instalace a aktualizací po konfiguraci registru, správu chyb a systémové správy. Licencování je klíčovým prvkem, který zajišťuje legální používání systému. Správa systému pomocí nástrojů, jako jsou Task Manager, Device Manager a bezpečnostní zásady, umožňuje udržet systém efektivní a bezpečný. Kvalitní nasazení a správná údržba zajišťují plynulý chod systému Windows a jeho bezpečnost pro uživatele.

23. Správa procesů a služeb ve MS Windows

správa procesů, komunikace mezi procesy, programové rozhraní, kompatibilita

23.1. Správa procesů ve Windows

Proces je instance spuštěného programu, která využívá systémové prostředky (CPU, paměť, I/O) pro svůj běh. Správa procesů zahrnuje vytváření, řízení a ukončování procesů.
Nástroje pro správu procesů

1. Správce úloh (Task Manager):
    - Poskytuje uživatelské rozhraní pro sledování a správu běžících procesů.
    - Zobrazuje informace o využití CPU, paměti, disku, sítě a dalších prostředků.
    - Umožňuje ukončit procesy, které neodpovídají nebo způsobují problémy.
2. Příkazový řádek (Command Prompt) a PowerShell:
    - **tasklist**: Příkaz zobrazí seznam běžících procesů v systému.
    - **taskkill**: Umožňuje ukončit proces podle jeho PID nebo názvu.
    - **Get-Process** (v PowerShellu): Zobrazuje informace o běžících procesech, včetně jejich ID a využití paměti.
    - **Stop-Process** (v PowerShellu): Ukončuje procesy pomocí jejich názvu nebo PID.
3. Správce úloh služby (Service Manager):
    - Umožňuje spravovat služby, které běží na pozadí systému Windows.
    - Poskytuje možnosti pro spuštění, zastavení, pozastavení nebo restartování služeb.

Životní cyklus procesu

1. **Vytvoření procesu**: Nový proces je vytvořen voláním funkce CreateProcess. Windows načte binární soubor programu, alokuje potřebné prostředky a inicializuje nový proces.
2. **Plánování a řízení procesu**: Systém Windows používá plánovač, který přiděluje čas procesoru jednotlivým procesům na základě priorit.
3. **Ukončení procesu**: Proces se ukončí buď úspěšně po dokončení, nebo je násilně ukončen (např. uživatelem ve Správci úloh).

Priorita procesu

- Každý proces má přiřazenou **prioritu** (např. Realtime, High, Normal, Low), která určuje, jak často bude plánovač CPU proces vyvolávat k běhu.
- Uživatel může měnit prioritu procesů pomocí Správce úloh nebo příkazem PowerShell.

23.2. Komunikace mezi procesy (Interprocess Communication, IPC)

Komunikace mezi procesy (IPC) umožňuje procesům vyměňovat si data a spolupracovat. Windows podporuje několik metod IPC pro synchronizaci a přenos dat mezi procesy.
Metody IPC ve Windows

1. Pojmenované roury (Named Pipes):
    - Poskytují obousměrnou komunikaci mezi procesy, které mohou běžet na stejném počítači nebo v síti.
    - Používají se často pro komunikaci mezi aplikacemi klient-server.
2. Anonymní roury (Anonymous Pipes):
    - Umožňují jednosměrnou komunikaci mezi dvěma procesy na stejném počítači.
    - Obvykle se využívají pro komunikaci mezi procesy, které mají vztah rodič-potomek.
3. Sdílená paměť (Shared Memory):
    - Umožňuje procesům sdílet část paměti, což poskytuje rychlý přístup k datům.
    - Vyžaduje synchronizaci pomocí semaforů nebo mutexů, aby nedocházelo ke kolizím.
4. Sokety (Sockets):
    - Umožňují komunikaci mezi procesy přes síť pomocí protokolů jako TCP/IP.
    - Často se využívají pro aplikace klient-server.
5. Signály a události (Events):
    - Pomocí signálů mohou procesy informovat jiné procesy o určitých událostech.
    - **Eventy (události)** umožňují procesům synchronizovat své činnosti nebo čekat na dokončení úkolu jiného procesu.
6. Message Passing (Zasílání zpráv):

- Procesy mohou používat Windows zprávy k předávání informací, obzvláště v prostředí s GUI.

Synchronizační objekty

1. **Mutexy**: Zajišťují, že k určitému zdroji přistupuje pouze jeden proces najednou.
2. **Semafory**: Umožňují omezený počet přístupů k určitému zdroji.
3. **Události (Events)**: Slouží ke signalizaci mezi procesy a synchronizaci činností.

23.3. Programové rozhraní (API)

Windows API poskytuje sadu funkcí a nástrojů, které mohou aplikace a procesy používat pro komunikaci se systémem a pro využívání jeho služeb.

Windows API

- **Windows API** je rozhraní, které umožňuje aplikacím volat funkce operačního systému, jako je správa souborů, správa paměti, zpracování textu a grafiky, nebo správa síťových služeb.
- Windows API je rozděleno do několika knihoven:
  - **Kernel32.dll**: Základní systémové funkce, jako je správa procesů a paměti.
  - **User32.dll**: Funkce pro práci s uživatelským rozhraním a ovládacími prvky.
  - **GDI32.dll**: Grafické rozhraní pro zobrazení 2D grafiky.
  - **Advapi32.dll**: Funkce pro správu bezpečnosti, oprávnění a registru.

Win32 API

- **Win32 API** je podmnožina Windows API, která umožňuje přístup k základním systémovým funkcím a spravuje služby na nízké úrovni.
- Vývojáři používají Win32 API pro aplikace s vysokými nároky na výkon nebo přístup k hardwarovým funkcím.

Rozhraní pro správu služeb a procesů

- API funkce jako CreateProcess, OpenProcess, TerminateProcess umožňují vývojářům programově řídit procesy.
- API funkce StartService, StopService slouží k ovládání služeb, které běží na pozadí systému.

23.4. Kompatibilita

Kompatibilita zajišťuje, že aplikace, které byly vytvořeny pro starší verze systému Windows, mohou běžet na novějších verzích OS. Windows poskytuje různé nástroje a režimy, které pomáhají s kompatibilitou aplikací.

Režim kompatibility

- **Režim kompatibility** umožňuje spustit aplikace v prostředí, které simuluje starší verzi Windows.
- Umožňuje uživateli zvolit nastavení, jako je spuštění v režimu starší verze Windows, změna rozlišení nebo omezení vizuálních efektů.

Virtualizace

- **Windows Virtual PC**: Nástroj, který umožňuje provoz starších verzí systému Windows jako virtuálních strojů na moderních zařízeních.
- **Hyper-V**: Podporuje virtualizaci a umožňuje běh různých operačních systémů na jednom fyzickém počítači, což usnadňuje testování kompatibility.

Kompatibilita aplikací 32bit a 64bit

- 64bitové verze Windows poskytují kompatibilitu s 32bitovými aplikacemi díky **WoW64** (Windows on Windows 64), což je emulační vrstva, která umožňuje spouštět 32bitové aplikace na 64bitových systémech.
- 32bitové aplikace však nemohou využívat plný potenciál 64bitových systémů, například přístup k většímu množství paměti.

Kompatibilita pomocí API

- Windows API zajišťuje kompatibilitu tím, že se stará o změny ve funkcích a zabezpečení, aby aplikace fungovaly i na různých verzích Windows.
- Vývojáři mohou používat API, které zůstává stabilní přes různé verze OS, což zajišťuje, že aplikace budou fungovat i po aktualizaci OS.

Nástroje pro správu kompatibility

1. **Application Compatibility Toolkit (ACT)**: Nástroj od Microsoftu, který pomáhá diagnostikovat a opravovat problémy s kompatibilitou.

2. **Kompatibilní balíčky (Compatibility Packs)**: Poskytují aktualizace pro aplikace, aby správně fungovaly na novějších verzích OS.

23.5. Závěr

Správa procesů a služeb ve Windows zajišťuje, že systém funguje efektivně a bezpečně. Komunikace mezi procesy umožňuje spolupráci aplikací a efektivní sdílení zdrojů. Windows API poskytuje rozhraní pro řízení systémových funkcí, což usnadňuje vývoj aplikací, které mohou využívat systémové prostředky a služby. Kompatibilita je klíčová pro zachování funkčnosti starších aplikací v nových verzích systému Windows. Správné využití nástrojů pro kompatibilitu, jako je režim kompatibility nebo virtualizace, zajišťuje, že uživatelé mohou nadále používat aplikace, které jsou pro ně důležité.

24. Nasazení OS Linux

start systému, běh a ukončení systému, správa služeb, chyby při běhu, instalace aplikací, logování provozu a log soubory, licencování

24.1. Start systému

Proces startu systému Linux zahrnuje několik fází, které vedou k úplnému spuštění operačního systému.
Fáze startu systému Linux

1. Inicializace firmwaru (BIOS/UEFI):
    - Po zapnutí počítače provede základní deska kontrolu hardwaru (POST) a předá řízení zavaděči operačního systému.
2. Zavaděč systému (Bootloader):
    - **GRUB (GRand Unified Bootloader)** nebo **LILO** (v dřívějších verzích) je zavaděč používaný pro načtení Linuxového jádra.
    - Umožňuje výběr různých jader a zavádění více operačních systémů.
3. Načítání jádra:
    - Jádro Linuxu se načte do paměti a inicializuje základní hardwarové ovladače.
    - Jádro následně inicializuje proces init (nebo systém systemd na moderních distribucích).
4. Init systém nebo systemd:
    - **Init** je tradiční inicializační systém, který spouští procesy definované v /etc/inittab.
    - **systemd** je moderní init systém využívaný na většině distribucí Linuxu, který spouští služby paralelně a rychleji než tradiční init.
5. Spuštění uživatelského prostředí:
    - Po startu základních služeb je spuštěno grafické prostředí (pokud je dostupné) a uživateli je umožněno přihlášení.

Řešení problémů se startem

- **Single User Mode**: Nouzový režim, který umožňuje spuštění systému s minimálními službami pro účely diagnostiky a opravy.
- **Obnova zavaděče GRUB**: Pokud dojde k poškození GRUB, lze jej obnovit pomocí live distribuce Linuxu a přístupem k příkazům grub-install a update-grub.

24.2. Běh a ukončení systému

Běh systému

- Linux během běhu spravuje systémové procesy, uživatelské prostředí a služby, které podporují síťové připojení, úložiště a další funkce.
- **Systemd** spravuje spuštění a zastavení jednotlivých služeb a provádí sledování systémového času a správy logů.

Ukončení systému

- Systém lze ukončit různými způsoby:
    - **shutdown**: Slouží k vypnutí systému, např. shutdown -h now vypne systém ihned.
    - **reboot**: Restartuje systém.
    - **halt**: Zastaví systém a připraví jej k vypnutí.
- Signály pro ukončení procesů:
    - SIGTERM (signál pro jemné ukončení) – umožňuje procesům dokončit probíhající úkoly.
    - SIGKILL (signál pro okamžité ukončení) – okamžitě ukončí proces, který nereaguje na SIGTERM.

24.3. Správa služeb

Služby (démoni) jsou procesy, které běží na pozadí a podporují různé funkce systému (např. síťové služby, úložiště). Správa služeb umožňuje kontrolovat, které služby jsou spuštěny a jaký je jejich stav.
Systemd

- **Systemd** je moderní správce služeb, který se používá na většině distribucí Linuxu.
- Hlavní příkazy:
    - systemctl start <název_služby>: Spustí službu.

- systemctl stop <název_služby>: Zastaví službu.
- systemctl restart <název_služby>: Restartuje službu.
- systemctl enable <název_služby>: Povolení služby při startu systému.
- systemctl status <název_služby>: Zobrazí stav služby.

Init systém

- V dřívějších distribucích byl používán systém **SysV init**, kde jsou služby spravovány skripty uloženými v /etc/init.d/.
- Služby lze spustit pomocí příkazů, jako je /etc/init.d/<název_služby> start.

24.4. Chyby při běhu

Chyby při běhu systému mohou být způsobeny aplikacemi, konflikty mezi procesy nebo problémy s hardwarem.
Typy chyb

1. **Jádrové chyby (Kernel Panics)**: Kritické chyby jádra, které způsobují okamžité zastavení systému.
2. **Chyby služeb**: Služby mohou selhat kvůli nesprávné konfiguraci nebo chybám v softwaru.
3. **Paměťové chyby**: Mohou být způsobeny nedostatkem volné paměti nebo konflikty mezi aplikacemi.

Diagnostika chyb

- **Prohlížení logů**: Logy obvykle poskytují informace o chybách, které se vyskytnou během běhu. Lze je prohlížet pomocí příkazu journalctl nebo v souborech uložených ve /var/log.
- **Safe Mode nebo Recovery Mode**: Umožňuje spuštění systému s minimálním nastavením pro diagnostiku.
- **Příkaz dmesg**: Zobrazuje log zpráv jádra, včetně hardwarových chyb.

24.5. Instalace aplikací

Instalace softwaru v Linuxu se provádí pomocí balíčkovacích systémů, které umožňují správu aplikací, jejich aktualizace a odstraňování.
Balíčkovací systémy

1. DEB balíčky (Debian-based):
   - Používá se na distribucích jako Ubuntu a Debian.
   - **APT (Advanced Package Tool)** je nástroj pro správu DEB balíčků.
   - Příkazy:
     - apt update: Aktualizuje seznam balíčků.
     - apt install <název_balíčku>: Instaluje balíček.
     - apt remove <název_balíčku>: Odstraní balíček.
2. RPM balíčky (Red Hat-based):
   - Používá se na distribucích jako CentOS, Fedora a RHEL.
   - **YUM** a **DNF** jsou nástroje pro správu RPM balíčků.
   - Příkazy:
     - yum install <název_balíčku> nebo dnf install <název_balíčku>.
     - yum update <název_balíčku> nebo dnf update <název_balíčku>.
3. Snap a Flatpak:
   - Univerzální balíčkové formáty, které fungují napříč distribucemi a umožňují instalaci aplikací se všemi závislostmi.
   - **Snap**: Používá Canonical (Ubuntu) – snap install <název_balíčku>.
   - **Flatpak**: Zajišťuje izolaci aplikací – flatpak install <název_balíčku>.

24.6. Logování provozu a log soubory

Logovací systém Linuxu zaznamenává různé události a umožňuje diagnostikovat chyby a sledovat činnost systému.
Hlavní logovací soubory a jejich umístění

- **/var/log/syslog** nebo **/var/log/messages**: Záznamy systémových událostí a zpráv.
- **/var/log/auth.log**: Informace o přihlášení a autorizaci uživatelů.
- **/var/log/dmesg**: Obsahuje zprávy jádra, které se generují při spuštění systému.
- **/var/log/kern.log**: Specifické záznamy o jádru systému.
- **/var/log/boot.log**: Záznamy z procesu startu systému.
- **/var/log/secure**: Obsahuje záznamy bezpečnostních událostí (u distribucí založených na Red Hat).

Nástroje pro práci s logy

- **journalctl**: Nástroj pro čtení logů systému spravovaných systemd. Příklady příkazů:
  - journalctl -b: Zobrazí logy od posledního startu systému.
  - journalctl -u <název_služby>: Zobrazí logy specifické služby.
- **rsyslog**: Systém pro správu a ukládání logů v textových souborech, běžně používaný na Linuxových systémech.

24.7. Licencování

Linux je licencován pod **GNU General Public License (GPL)**, což je otevřená licence umožňující volné používání, modifikaci a distribuci softwaru.
Klíčové principy GPL licence

- **Svoboda používání**: Uživatelé mohou Linux používat pro jakýkoliv účel.
- **Přístup ke zdrojovému kódu**: Zdrojový kód je k dispozici všem uživatelům, což umožňuje jeho modifikaci.
- **Redistribuce**: Uživatelé mohou upravený nebo původní kód znovu distribuovat.
- **Zachování licenčních práv**: Každý upravený nebo redistribuovaný software musí být nadále dostupný pod GPL licencí.

Další licence používané v Linuxu

- **LGPL (Lesser GPL)**: Používá se pro knihovny, které lze používat i v proprietárních aplikacích.
- **Apache License** a **MIT License**: Otevřené licence umožňující volnější podmínky pro komerční využití.

24.8. Závěr

Nasazení a správa systému Linux zahrnují jeho start, běh, ukončení a správu služeb, instalaci aplikací a diagnostiku chyb. Logovací systém zajišťuje sledování činností a identifikaci problémů. Správa balíčků umožňuje snadnou instalaci a aktualizaci aplikací. GPL licence poskytuje otevřenost, flexibilitu a zajišťuje, že Linux je dostupný všem uživatelům. Tento soubor nástrojů a procesů zajišťuje stabilní, bezpečné a efektivní využívání systému Linux.

25. Správa procesů a služeb OS Linux

skupiny a relace procesů, plánování procesů, příkazy pro práci s procesy, úlohy a multitasking

25.1. Skupiny a relace procesů

Linux umožňuje organizovat procesy do **skupin** a **relací procesů**, což usnadňuje správu a kontrolu nad více procesy, zejména pokud jsou propojeny hierarchicky.
Skupiny procesů

- **Skupina procesů (Process Group)**: Kolekce procesů, které sdílejí stejný identifikátor skupiny procesů (PGID). Každý proces má svůj PID (Process ID), ale procesy v jedné skupině mají stejný PGID.
- Skupiny procesů se používají k řízení přístupu k terminálu a odesílání signálů. Například signál SIGKILL lze poslat celé skupině procesů.

Relace procesů

- **Relace procesů (Process Session)**: Skupina procesů, které sdílejí stejnou relaci. Relace může zahrnovat jednu nebo více skupin procesů a má jednoho **vedoucího procesu relace**.
- Relace procesů se často používají v příkazových interpretech, kde je každá relace svázána s uživatelskou relací (přihlášení do systému) a zahrnuje všechny procesy zahájené během této relace.
- Terminály a příkazové řádky připojují relace procesů, takže například po ukončení relace (odhlášení uživatele) jsou ukončeny všechny procesy, které do ní patří.

Příkazy pro práci se skupinami a relacemi procesů

- **pgrep**: Umožňuje vyhledávat procesy podle jejich PGID nebo názvu procesu (např. pgrep -g <PGID> vyhledá procesy ve skupině).
- **pkill**: Umožňuje posílat signály procesům v rámci skupiny (např. pkill -g <PGID> ukončí všechny procesy ve skupině).
- **ps**: Zobrazuje informace o běžících procesech, včetně jejich PID, PGID, PPID (Parent PID) a dalších atributů.

25.2. Plánování procesů

Plánování procesů je základní součástí správy procesů, která zajišťuje, že systém optimálně přiděluje CPU jednotlivým procesům na základě jejich priorit a potřeb.
Plánovací algoritmy v Linuxu
Linux využívá **plánovač CFS (Completely Fair Scheduler)**, který poskytuje efektivní a vyvážené přidělování procesoru:

1. Completely Fair Scheduler (CFS):
    - CFS funguje tak, aby každý proces dostal spravedlivý podíl času na CPU. Sleduje čas strávený každým procesem na procesoru a zajišťuje, že žádný proces nebude „hladovět“.
    - CFS používá koncept **virtuálního runtime**, který zohledňuje prioritu procesů a umožňuje spravedlivé plánování i pro systémy s různými požadavky na prioritu.
2. Priority procesů:
    - Linux používá rozsah priorit od -20 (nejvyšší priorita) po +19 (nejnižší priorita).
    - Priorita procesu je určena atributem **niceness** (nice), který nastavuje relativní prioritu procesu. Výchozí hodnota je 0, vyšší hodnoty znamenají nižší prioritu.
3. Reálný čas (Real-Time Scheduling):
    - Linux podporuje i plánování pro reálný čas pomocí algoritmů **SCHED_FIFO** a **SCHED_RR**, které zajišťují, že kritické procesy mají vyšší prioritu a přednostní přístup k CPU.

Nastavení priority a plánování

- **nice <priority> <program>**: Spustí program s nastavenou hodnotou nice.
- **renice <priority> <PID>**: Změní hodnotu nice u existujícího procesu podle jeho PID.
- **chrt**: Příkaz umožňující specifikovat plánovací politiku (např. FIFO, RR) pro procesy v reálném čase.

25.3. Příkazy pro práci s procesy

Linux poskytuje mnoho příkazů, které umožňují sledovat, spravovat a řídit běžící procesy. Mezi nejdůležitější patří:

1. ps:

- Zobrazuje seznam běžících procesů s informacemi, jako je PID, PPID, využití paměti, uživatel a stav procesu.
- Například ps aux zobrazí všechny procesy s podrobnými informacemi o uživatelích.

2. top:
   - Interaktivní nástroj pro monitorování procesů v reálném čase, který zobrazuje využití CPU, paměti a další metriky.
   - Umožňuje řazení procesů podle různých kritérií (např. spotřeba paměti, zatížení CPU).
3. htop:
   - Rozšířená verze top s intuitivnějším rozhraním a většími možnostmi přizpůsobení. Zobrazuje procesy v barevném rozhraní a umožňuje snadnou navigaci.
4. kill:
   - Posílá signály procesům podle jejich PID.
   - kill -9 <PID> nebo kill -SIGKILL <PID>: Posílá signál SIGKILL pro okamžité ukončení procesu.
   - kill -15 <PID> nebo kill -SIGTERM <PID>: Jemnější ukončení procesu.
5. pkill a pgrep:
   - pgrep <název_procesu>: Vyhledá procesy podle názvu.
   - pkill <název_procesu>: Ukončí procesy podle názvu.
6. bg a fg:
   - bg: Přesune proces na pozadí, takže běží v režimu pozadí.
   - fg: Přesune proces zpět do popředí.
7. jobs:
   - Zobrazuje seznam aktuálních úloh, které jsou spuštěné nebo pozastavené v rámci terminálu.

25.4. Úlohy a multitasking

Linux je operační systém, který podporuje **multitasking** – umožňuje běh více procesů současně.
Úlohy a jejich správa

- **Úloha (Job)**: Proces nebo sada procesů, které jsou spuštěny v rámci jedné příkazové relace.
- Pozadí a popředí:
  - Proces lze spustit na pozadí (background) přidáním & za příkaz (např. sleep 10 &).
  - Proces na pozadí neblokuje terminál a umožňuje uživateli zadávat další příkazy.
  - Pomocí příkazů fg a bg lze přesouvat úlohy mezi pozadím a popředím.

Multitasking a plánování úloh

1. Preemptivní multitasking:
   - Linux používá preemptivní multitasking, což znamená, že procesy jsou přerušovány operačním systémem, aby jiným procesům bylo umožněno získat čas na CPU.
   - Plánovač procesu vybere další proces na základě priorit a dostupného času.
2. Daemon procesy:
   - **Démoni** jsou procesy běžící na pozadí, které obvykle nemají přímou interakci s uživatelem a běží po celou dobu chodu systému (např. síťové služby, plánovače).
   - Tyto procesy se spouští automaticky při startu systému a jsou spravovány systemd nebo jiným init systémem.
3. Plánovač úloh (Cron):
   - Cron umožňuje nastavit úlohy, které se mají spouštět automaticky v pravidelných intervalech.
   - Konfigurace úloh se provádí v souborech crontab.
   - **Syntaxe crontab**: Minuta, hodina, den v měsíci, měsíc, den v týdnu, příkaz (např. 0 3 * * * /home/user/backup.sh pro spuštění denní zálohy ve 3:00 ráno).
4. at:
   - Jednorázový plánovač úloh, který umožňuje spustit příkaz v určený čas.
   - **Syntaxe**: at <čas> a následné zadání příkazu, který má být spuštěn (např. at now + 5 minutes).

25.5. Závěr

Správa procesů a služeb v Linuxu zahrnuje organizaci procesů do skupin a relací, plánování pomocí CFS a reálného času, a efektivní multitasking, který umožňuje běh více procesů najednou. Příkazy jako ps, top, kill a systemctl poskytují uživatelům a správcům systémů nástroje pro monitorování a řízení procesů. Podpora multitaskingu umožňuje správné rozložení

systémových prostředků, zatímco plánovače úloh (cron, at) umožňují efektivní automatizaci rutinních činností. Tyto mechanismy zajišťují spolehlivý a efektivní běh systému Linux.

26. Zabezpečení OS

aktualizace systému, malware, nástroje proti malware, metody zabezpečení, problematika licencí

26.1. Aktualizace systému

Aktualizace operačního systému jsou klíčovým prvkem zabezpečení, jelikož poskytují opravy chyb, zabezpečení proti novým hrozbám a zajišťují stabilitu systému.
Typy aktualizací

1. Bezpečnostní aktualizace:
    - Opravy chyb, které představují bezpečnostní rizika, jako jsou zranitelnosti využitelné pro kybernetické útoky.
    - Zahrnují záplaty pro jádro, aplikace a knihovny systému.
2. Kumulativní aktualizace:
    - Obsahují všechny dosavadní aktualizace, což umožňuje rychlou instalaci všech oprav najednou.
3. Funkční aktualizace:
    - Rozšiřují systém o nové funkce nebo vylepšení, které mohou zahrnovat i nová bezpečnostní opatření.
4. Volitelné aktualizace:
    - Zahrnují vylepšení nebo nové verze ovladačů, které nejsou nezbytné, ale mohou zlepšit výkon nebo stabilitu.

Automatizace aktualizací

- **Automatické aktualizace**: Moderní operační systémy, jako Windows, macOS nebo Linux (včetně správců balíčků jako APT nebo YUM), umožňují automatickou instalaci aktualizací, což snižuje riziko zastaralosti systému.
- **Manuální aktualizace**: Správci systémů mohou řídit, kdy se aktualizace instalují, aby zabránili přerušení provozu. To je běžné ve firemních prostředích, kde aktualizace mohou být schváleny po testování.

Význam aktualizací

- Udržování systému aktuální je zásadní pro prevenci zneužití známých zranitelností, ochranu proti exploitům a zabezpečení dat v systému.

26.2. Malware

Malware (malicious software) zahrnuje škodlivé programy, které mohou poškodit systém, získat neoprávněný přístup nebo krást data.
Typy malwaru

1. Viry:
    - Škodlivý kód, který se připojuje k hostitelskému programu a šíří se infikací jiných souborů nebo systémů.
2. Červi (Worms):
    - Šíří se samostatně přes sítě a mohou infikovat další zařízení bez potřeby hostitelského programu.
3. Trojské koně (Trojans):
    - Škodlivý software, který se vydává za užitečný program. Po spuštění však provádí škodlivé činnosti, jako je získávání neoprávněného přístupu.
4. Ransomware:
    - Šifruje data na zařízení a vyžaduje výkupné za jejich dešifrování. Často se šíří prostřednictvím příloh e-mailů nebo nedůvěryhodných odkazů.
5. Spyware a adware:
    - **Spyware** sleduje aktivity uživatele a shromažďuje citlivé informace (např. hesla).
    - **Adware** zobrazují nechtěnou reklamu nebo shromažďuje data o uživatelském chování.
6. Rootkity:
    - Software, který se ukrývá v systému a umožňuje útočníkovi plný přístup k infikovanému zařízení. Rootkity mohou být obtížně detekovatelné a odstraňovatelné.

Způsoby infekce

- **E-maily a phishing**: Škodlivé přílohy nebo odkazy, které vedou k infekci systému.
- **Infikované webové stránky**: Návštěva nedůvěryhodné stránky může vést k neúmyslnému stažení malwaru.

- **USB zařízení**: Přenosný malware může být rozšířen prostřednictvím infikovaných USB disků.

26.3. Nástroje proti malware

Operační systémy i specializovaný software poskytují nástroje pro ochranu proti malwaru, jeho detekci a odstranění.
Antivirové a antimalwarové programy

- **Antiviry**: Software určený k detekci, blokování a odstranění malwaru. Mezi oblíbené antiviry patří Windows Defender, Avast, Kaspersky a další.
- **Antimalwarové nástroje**: Nástroje zaměřené na specifické druhy malwaru, například Malwarebytes, které se zaměřují na spyware, adware a trojské koně.

Firewally

- **Softwarové firewally**: Monitorují síťovou aktivitu a blokují neautorizované připojení.
- **Hardwarové firewally**: Poskytují bezpečnostní bariéru na úrovni sítě a mohou filtrovat připojení, která nesplňují bezpečnostní pravidla.

Skenery a analyzátory

- **Skenery malware**: Pravidelně kontrolují systém na přítomnost malwaru a mohou být nastaveny na automatické nebo plánované skenování.
- **IDS/IPS (Intrusion Detection and Prevention Systems)**: Nástroje monitorující síťovou aktivitu, detekují a zabraňují neoprávněným přístupům nebo pokusům o útok.

Prevence a ochranná opatření

- **Sandboxing**: Umožňuje spustit podezřelé programy v izolovaném prostředí, kde nemohou poškodit systém.
- **Předběžné ověřování**: Funkce v moderních OS, která kontroluje bezpečnost aplikací ještě před jejich spuštěním (např. Windows SmartScreen, AppArmor v Linuxu).

26.4. Metody zabezpečení

Efektivní zabezpečení operačního systému vyžaduje několik vrstev ochrany a postupů, které minimalizují rizika.
Metody zabezpečení

1. Řízení přístupu a oprávnění:
    - Nastavení uživatelských účtů s minimálními oprávněními podle zásady nejmenší nutné oprávnění.
    - Používání silných hesel a vícefaktorové autentizace (MFA).
2. Šifrování:
    - **Šifrování disku**: Umožňuje zabezpečit data na disku před neoprávněným přístupem (např. BitLocker ve Windows, LUKS v Linuxu).
    - **Šifrování komunikace**: Použití protokolů jako TLS pro zabezpečení dat přenášených po síti.
3. Pravidelné zálohování:
    - Zálohy jsou nezbytné pro obnovu systému v případě útoku nebo selhání. Nejlepší praxí je využití automatizovaných záloh a jejich uložení na odděleném místě.
4. Monitorování a logování:
    - Pravidelná kontrola logů pro zjištění podezřelé aktivity.
    - Nástroje pro monitorování integrity souborů, které detekují změny v systému (např. Tripwire).
5. Řízení síťových připojení:
    - Omezení přístupu k síťovým službám a zavedení pravidel pro příchozí a odchozí síťový provoz.
    - Použití VPN k šifrování síťového provozu pro ochranu dat na veřejných sítích.
6. Předběžné ověřování aplikací (App Whitelisting):
    - Povolení pouze důvěryhodných aplikací k běhu na systému, čímž se minimalizuje riziko spuštění škodlivého kódu.
7. Pravidelná bezpečnostní školení:
    - Uživatelé by měli být školeni na rozpoznání phishingových útoků a bezpečnostních hrozeb, aby mohli sami přispět k ochraně systému.

26.5. Problematika licencí

Licencování softwaru je zásadní pro legální používání OS a aplikací a také pro jejich bezpečnost, protože legálně licencovaný software obvykle obsahuje pravidelné aktualizace a podporu.
Typy licencí

1. Proprietární licence:
    - Umožňuje používání softwaru podle podmínek stanovených výrobcem, např. Windows, Microsoft Office.
    - Proprietární software je obvykle uzavřený a nedovoluje uživatelům měnit zdrojový kód.
2. Open-source licence:
    - Zdrojový kód je k dispozici veřejnosti a uživatelé jej mohou upravovat a distribuovat (např. licence GPL pro Linux).
    - Open-source licence podporuje bezpečnost prostřednictvím transparentnosti a komunitního přístupu.
3. Freeware:
    - Software, který je zdarma k používání, ale zdrojový kód není obvykle k dispozici (např. Adobe Reader).
4. Shareware:
    - Software, který je dostupný zdarma pro vyzkoušení po omezenou dobu, poté je nutná platba.

Význam legální licence

- **Pravidelné aktualizace**: Legálně licencovaný software dostává bezpečnostní záplaty a aktualizace, což je důležité pro ochranu systému.
- **Podpora výrobce**: Držitelé legálních licencí mají přístup k technické podpoře a dokumentaci.
- **Právní ochrana**: Použití legálního softwaru eliminuje riziko právních postihů za porušení autorských práv.
- **Bezpečnost**: Nelegální software často obsahuje malware nebo zadní vrátka (backdoors), která mohou být použita k útoku na systém.

26.6. Závěr

Zabezpečení operačního systému vyžaduje komplexní přístup zahrnující pravidelné aktualizace, nástroje proti malwaru, efektivní metody řízení přístupu a uživatelská školení. Různé typy licencí poskytují právní rámec a určují dostupnost aktualizací a podpory. Pravidelné sledování bezpečnostních protokolů a správa uživatelských oprávnění pomáhají udržet operační systém odolný vůči útokům a zneužití.