

1. Elektronika – Základy číslicovej techniky - Samo

a)

1. Charakterizujte význam binárneho číselného kódu vo výpočtovej technike
2. Porovnajte kombinačné a sekvenčné logické obvody
3. Porovnajte pravdivostné tabuľky dvojjstupových logických obvodov AND, NAND, OR a NOR
4. Objasnite rozdiel medzi monostabilným (M.K.O.), bistabilným (B.K.O.) a astabilným klopným obvodom (A.K.O.)
5. Vysvetlite a popíšte reprodukčný proces podniku, jeho vstupy a výstupy

b)

1. Definujte hexadecimálnu sústavu
2. Načrtnite využitie hexadecimálneho kódu v počítačových sieťach
3. Charakterizujte binárnu číselnú sústavu a jej využitie v praxi

c)

1. Zvoľte si a vyjadrite v binárnom kóde IPv4 adresu s dĺžkou prefixu 24
2. Porovnajte IPv4 siete a IPv4 hosta

a1 Charakterizujte význam binárneho číselného kódu vo výpočtovej technike

Základná jednotka pre ukladanie a manipuláciu s informáciami: **Binarný kód** používa dve hodnoty (0 a 1) na reprezentáciu všetkých údajov. Táto jednoduchosť je kľúčová pre digitálnu technológiu, pretože počítače môžu pracovať s týmito dátami veľmi efektívne.

Digitalizácia informácií: Vďaka binárnemu kódu môžeme premeniť analógové údaje na digitálnu formu. To je dôležité pre uchovávanie, prenos a spracovanie informácií v počítačoch a digitálnych zariadeniach.

Jednoduchosť logických operácií: Logické operácie ako AND, OR a NOT sú jednoducho implementovateľné pomocou binárneho kódu. To umožňuje počítačom vykonávať zložité úlohy, ako sú napríklad výpočty alebo spracovanie dát.

Komunikácia a prenos dát: Vďaka binárnemu kódu môžeme prenášať dáta medzi rôznymi digitálnymi zariadeniami a počítačmi. Tento kód je štandardizovaný, čo umožňuje interoperabilitu medzi rôznymi systémami.

Dátové šifrovanie a bezpečnosť: Binarný kód je základom pre dátové šifrovanie, ktoré zabezpečuje bezpečnú komunikáciu a uchovávanie informácií.

a2 Porovnajte kombinačné a sekvenčné logické obvody

Kombinačné logické obvody:

Charakteristika: Kombinačné logické obvody generujú výstup na základe okamžitého stavu vstupov. Výstup závisí výlučne na aktuálnych hodnotách vstupov a funkcii, ktorá je implementovaná v obvode.

Príklady: Logické brány (AND, OR, NOT, XOR), multiplexory, dekodéry, komparátory atď.

Využitie: Používajú sa na implementáciu jednoduchých aj zložitých logických funkcií, ktoré nemajú vnútornú pamäť alebo stav.

Sekvenčné logické obvody:

Charakteristika: Sekvenčné logické obvody majú pamäťový prvok, ktorý uchováva stav a mení svoj výstup v čase na základe aktuálnych vstupov a stavu pamäte.

Príklady: Pravdepodobne najznámejším príkladom je sekvenčný obvod s klopňým obvodom, napríklad D klopňý obvod alebo JK klopňý obvod.

Využitie: Používajú sa na uchovávanie informácií a implementáciu časovo závislých funkcií. Často sa využívajú v sekvenčných logických systémoch, ako sú čítače, registre, stavové automaty a procesory.

Porovnanie:

Časová závislosť: Kombinačné obvody nemajú vnútorný stav a generujú výstup okamžite, zatiaľ čo sekvenčné obvody majú pamäť a ich výstup závisí aj od predchádzajúceho stavu.

Pamäťové prvky: Kombinačné obvody nemajú pamäťové prvky, zatiaľ čo sekvenčné obvody ich majú a môžu uchovávať informácie medzi jednotlivými vstupmi.

Implementácia funkcií: Kombinačné obvody implementujú funkcie bez pamäte, zatiaľ čo sekvenčné obvody môžu implementovať funkcie s pamäťou a časovou závislosťou.

V praxi sa kombinuje použitie oboch typov obvodov pre dosiahnutie požadovaných funkcií a výkonu digitálnych systémov.

a3 Porovnajete pravdivostné tabuľky dvojjstupových logických obvodov AND, NAND, OR a NOR

Pravdivostné tabuľky slúžia k popisu výstupov logických obvodov v závislosti na vstupoch. Tu sú pravdivostné tabuľky pre dvojjstupové logické obvody AND, NAND, OR a NOR:

AND (Logický súčin):

A	B	Výstup
0	0	0
0	1	0
1	0	0
1	1	1

Výstup je 1 iba vtedy, ak sú oba vstupy A aj B rovnaké 1.

NAND (Negovaný AND):

A	B	Výstup
---	---	--------

0	0	1
0	1	1
1	0	1
1	1	0

Výstup je 0 iba vtedy, ak sú oba vstupy A aj B rovnaké 1, inak je výstup 1.

OR (Logický súčet):

A	B	Výstup
0	0	0
0	1	1
1	0	1
1	1	1

Výstup je 1, ak je aspoň jeden z vstupov A alebo B rovnaký 1.

NOR (Negovaný OR):

A	B	Výstup
0	0	1
0	1	0
1	0	0
1	1	0

Výstup je 0, ak je aspoň jeden z vstupov A alebo B rovnaký 1, inak je výstup 1.

Porovnanie:

AND a NAND: Sú si navzájom komplementárne. Výstup NAND je inverzný k výstupu AND.

OR a NOR: Podobne sú si navzájom komplementárne. Výstup NOR je inverzný k výstupu OR.

AND a OR: Sú základnými logickými operáciami a poskytujú rôzne funkcionality.

NAND a NOR: Sú základnými logickými operáciami, ktoré sa dajú použiť na implementáciu akýchkoľvek ďalších logických operácií.

Tieto pravdivostné tabuľky sú základom pre analýzu a návrh logických obvodov v digitálnej elektronike.

a4 Objasnite rozdiel medzi monostabilným (M.K.O.), bistabilným (B.K.O.) a astabilným klopným obvodom (A.K.O.)

Klopné obvody sú elektronické obvody, ktoré majú schopnosť udržiavať jeden zo svojich stavov (0 alebo 1) aj po zmene vstupných podmienok. Existujú tri hlavné typy klopných obvodov: monostabilné, bistabilné a astabilné. Tu je objasnenie rozdielov medzi nimi:

Monostabilný klopný obvod:

Charakteristika: Monostabilný klopný obvod má jediný stabilný stav. Po prijatí vstupného signálu vstúpi do nestabilného stavu počas určitého času (označovaného ako časová konštanta), po ktorom sa vráti do svojho stabilného stavu.

Využitie: Používa sa na generovanie krátkych impulzových signálov alebo oneskorení.

Bistabilný klopný obvod:

Charakteristika: Bistabilný klopný obvod má dva stabilné stavy a môže zostať v ktoromkoľvek z nich aj po zmene vstupných podmienok. Nie je závislý na časových konštantách a zostane v aktuálnom stave, kým nedostane inštrukciu na zmenu.

Využitie: Používa sa na uchovávanie pamäte alebo na realizáciu spínačov a prepínačov.

Astabilný klopný obvod:

Charakteristika: Astabilný klopný obvod nemá stabilný stav. Neustále mení svoj stav medzi dvoma rôznymi hodnotami v pravidelných intervaloch, čo vytvára periodický signál.

Využitie: Používa sa na generovanie oscilačných signálov, ako sú kvapalné hodiny, blikajúce svetlá alebo generátory pulzov.

Rozdiel medzi nimi spočíva v ich schopnosti udržiavať stav a v ich účele. Monostabilné klopné obvody generujú krátke impulzné signály, bistabilné klopné obvody uchovávajú pamäť, zatiaľ čo astabilné klopné obvody generujú periodické signály. Každý z týchto typov má svoje vlastné využitie a aplikácie v elektronike.

a5 Vysvetlite a popíšte reprodukčný proces podniku, jeho vstupy a výstupy

Reprodukčný proces:**Vstupy:**

Ľudský kapitál: Zamestnanci, manažéri a ďalší personál, ktorí prispievajú svojimi schopnosťami a znalosťami k vykonávaniu práce.

Fyzické zdroje: Materiály, suroviny, vybavenie a technológie potrebné na výrobu produktov alebo poskytovanie služieb.

Finančné zdroje: Kapitál potrebný na prevádzku podniku, vrátane investícií, pôžičiek, príjmov a nákladov.

Proces spracovania:

Výrobné procesy: Manipulácia s materiálmi, spracovanie, montáž, testovanie a výroba produktov.

Poskytovanie služieb: Vyhodnocovanie požiadaviek zákazníkov, poskytovanie služieb zákazníkovi a riadenie vzťahov so zákazníkmi.

Výstupy:

Produkty: Fyzické výstupy procesu, ktoré sú určené na predaj alebo distribúciu na trhu.

Služby: Nehmotné výstupy procesu, ktoré poskytujú hodnotu zákazníkom a môžu zahŕňať poradenstvo, opravy, údržbu atď.

Zisk a hodnota pre zainteresované strany: Ekonomický zisk, spokojnosť zákazníkov, kvalita výrobkov a služieb a podpora rastu a udržateľnosti podniku.

Príklad:

Predstavme si výrobný podnik, ktorý vyrába nábytok. Jeho reprodukčný proces by mohol vyzeráť nasledovne:

Vstupy:

Ľudský kapitál: Zruční remeselníci, dizajnéri, manažéri.

Fyzické zdroje: Drevo, kovové súčiastky, farby, náradie.

Finančné zdroje: Investície na nákup surovín a vybavenia, prevádzkové náklady.

Proces spracovania:

Výrobné procesy: Rezné práce, tvarovanie, lepenie, farbenie, lakovanie.

Kvalita kontrola: Testovanie výrobkov na kvalitu a estetiku.

Výstupy:

Produkty: Rôzne druhy nábytku ako stoly, stoličky, skrine.

Služby: Dodanie a montáž nábytku u zákazníkov.

Zisk a hodnota pre zainteresované strany: Zisk z predaja, spokojnosť zákazníkov, dlhodobé zákaznícke vzťahy.

b1 Definujte hexadecimálnu sústavu

Hexadecimálna sústava, často označovaná ako hex, je pozíciový číselný systém s bázou 16. To znamená, že používa 16 rôznych symbolov na reprezentáciu číselných hodnôt. Hexadecimálna sústava je kombináciou desiatkovej sústavy (báza 10) a binárnej sústavy (báza 2).

Hexadecimálna sústava používa nasledujúce symboly:

Čísla od 0 do 9.

Až F (alebo f) pre reprezentáciu desiatkových hodnôt 10 až 15.

Pretože každý symbol môže reprezentovať 16 rôznych hodnôt, hexadecimálna sústava umožňuje kompaktné zápisy veľkých binárnych čísel. Každý symbol hexadecimálneho čísla reprezentuje 4 bity binárneho čísla.

b2 Načrtnite využitie hexadecimálneho kódu v počítačových sieťach

IP adresy: IP adresy, ktoré sú používané na identifikáciu zariadení v počítačovej sieti, sa bežne zapisujú pomocou štvorice oktetov v desiatkovej sústave oddelených bodkami. Napríklad IP adresa 192.168.1.1 sa môže zapisovať ako C0A80101 v hexadecimálnej sústave.

MAC adresy: MAC adresy, ktoré sú priradené sieťovým adaptérom zariadení a slúžia na ich jedinečnú identifikáciu v sieti, sú tiež zapisované pomocou hexadecimálnej sústavy. Každá časť MAC adresy reprezentuje 8 bitov, ktoré sa môžu zapisovať pomocou dvojčiferných hexadecimálnych čísel. Napríklad MAC adresa 00:1A:2B:3C:4D:5E by sa mohla zapisovať ako 001A2B3C4D5E.

Portové čísla: V TCP/IP protokole sú portové čísla, ktoré určujú koncové body komunikácie v sieti, bežne reprezentované ako 16-bitové čísla, ktoré sa môžu zapisovať pomocou hexadecimálnej sústavy. Napríklad port 80, ktorý je často používaný pre HTTP komunikáciu, sa zapisuje ako 0x50.

Šifrovacie kľúče: Pri zabezpečení komunikácie v počítačových sieťach sa často používajú šifrovacie kľúče, ktoré sú zvyčajne reprezentované ako dlhé reťazce hexadecimálnych čísel. Tieto kľúče môžu byť použité na šifrovanie a dešifrovanie dát v bezpečnej komunikácii.

b3 Charakterizujte binárnu číselnú sústavu a jej využitie v praxi

Dvojčísle: Binárna sústava používa len dva symboly - 0 a 1. Tieto symboly predstavujú dva stavy alebo hodnoty, často interpretované ako vypnuté (0) alebo zapnuté (1).

Pozíciový systém: V binárnej sústave má každá pozícia hodnotu, ktorá je mocninou čísla 2. Hodnota každej pozície sa zvyšuje o dvojnásobok predchádzajúcej pozície.

Jednotka informácie: Binárna sústava je vhodná na reprezentáciu informácií vo forme bitov, kde jeden bit reprezentuje jednu informačnú jednotku (0 alebo 1).

Jednoduché logické operácie: Logické operácie ako AND, OR a NOT sú jednoducho implementovateľné v binárnej sústave, čo umožňuje efektívne spracovanie informácií v digitálnych obvodoch a počítačoch.

Využitie v praxi:

Počítačové systémy: V počítačoch sa všetky údaje a inštrukcie reprezentujú pomocou binárnej sústavy. Procesory a ďalšie časti počítačových systémov manipulujú s binárnymi dátami pri vykonávaní operácií.

Digitálne komunikácie: Pri prenose dát cez digitálne komunikačné siete sa využíva binárna sústava na reprezentáciu informácií. Dáta sa prenášajú vo forme binárnych signálov (napr. výstupy z prístrojov alebo komunikačných zariadení).

Elektronika a robotika: V elektronike sa binárna sústava používa na návrh a analýzu digitálnych obvodov, ako aj na programovanie mikrokontrolérov a riadiacich systémov v robotike.

Kryptografia: V kryptografii sa binárna sústava používa na šifrovanie a dešifrovanie dát pomocou matematických operácií s binárnymi číslami.

c1 Zvoľte si a vyjadrite v binárnom kóde IPv4 adresu s dĺžkou prefixu 24

IPv4 adresa s dĺžkou prefixu 24 sa skladá z prvých 24 bitov, ktoré určujú sieťovú adresu, a zvyšných 8 bitov, ktoré určujú adresu zariadenia v tejto sieti. Tu je príklad binárneho zápisu takto zadanej IP adresy:

IPv4 adresa: 192.168.1.0/24

Binárna reprezentácia prvej časti (sieťová adresa) 192.168.1.0:

11000000 10101000 00000001 00000000

V tomto prípade prvých 24 bitov predstavuje sieťovú adresu a posledných 8 bitov bude nula, pretože je to sieťová adresa a nie konkrétne zariadenie v sieti.

c2 Porovnajete IPv4 siete a IPv4 hosta

IPv4 sieť:

Definícia: IPv4 sieť je skupina zariadení, ktoré sa nachádzajú v rovnakej logicky oddelovanej podsieti. Každá sieť má svoju unikátnu sieťovú adresu a dĺžku prefixu, ktorá určuje, koľko bitov v IP adrese je vyhradených pre sieťovú adresu.

Charakteristika: Sieťová adresa identifikuje konkrétnu sieť, ktorá môže obsahovať množstvo zariadení. Zvyčajne sa prvá časť IP adresy (sieťová adresa) používa na identifikáciu siete a zvyšok adresy (hostovská časť) na identifikáciu konkrétnych zariadení v danej sieti.

Príklad: Ak máme sieť s IP adresou 192.168.1.0/24, prvých 24 bitov (192.168.1) je sieťová adresa, kým zvyšných 8 bitov (0) identifikuje konkrétnu IP adresu zariadenia v tejto sieti.

IPv4 host:

Definícia: IPv4 host je konkrétne zariadenie (počítač, tlačiareň, server atď.), ktoré je súčasťou IPv4 siete. Každý host má svoju vlastnú jedinečnú IP adresu, ktorá ho jednoznačne identifikuje v rámci danej siete.

Charakteristika: Hostovská adresa je časť IP adresy, ktorá identifikuje konkrétny host v danej sieti. V rámci jednej siete musí byť adresa hosta jedinečná.

Príklad: V rámci siete s IP adresou 192.168.1.0/24 môže byť konkrétny host identifikovaný pomocou posledného bytu (oktetu) IP adresy. Napríklad zariadenie s IP adresou 192.168.1.10 je konkrétny host v tejto sieti.

Porovnanie:

Identifikácia: Sieťová adresa identifikuje sieť, zatiaľ čo hostovská adresa identifikuje konkrétny host v danej sieti.

Použitie prefixu: Sieťová adresa používa prefix na určenie dĺžky siete, zatiaľ čo hostovská adresa používa zvyšné bity na identifikáciu konkrétneho hosta.

Rozsah adres: Sieťová adresa má väčší rozsah (viacero adres v sieti), zatiaľ čo hostovská adresa je jedináčik v rámci danej siete.

2. Elektronika – Digitalizácia signálu a spracovanie informácií - Samo

a)

1. Nakreslite a popíšte rozdiely medzi analógovým, číslicovým a digitálnym signálom
2. Objasnite význam digitalizácie informácií
3. Popíšte časové priebehy na vstupoch a výstupoch časovačov TON a TOF

4. Popíšte základné kroky potrebné na digitalizáciu analógového signálu
5. Vysvetlite, aké finančné zdroje môže využívať výrobca PC a zaradte PC do majetku podniku

b)

1. Popíšte PDU na L2
2. Popíšte spôsoby komunikácie half duplex a full duplex
3. Vysvetlite spôsob komunikácie na spoločnom médiu

c)

1. Popíšte MAC tabuľku
2. Popíšte manažovanie switchu na L2

a1 Nakreslite a popíšte rozdiely medzi analógovým, číslicovým a digitálnym signálom

Analógový signál:

Reprezentácia: Analógový signál je spojitý signál, ktorý môže nadobúdať ľubovoľné hodnoty v určenom rozsahu. V každom časovom okamihu prenáša informáciu.

Charakteristika: Jeho hodnota môže postupne meniť svoju hodnotu a zahŕňa nekonečne veľa možných hodnôt v určenom rozsahu.

Grafické znázornenie: Krivka analógového signálu sa môže zobrazíť ako spojitá krivka, ktorá zvyčajne nevykazuje žiadne ostré zlomy alebo diskontinuity.

Číslicový signál:

Reprezentácia: Číslicový signál je diskretný signál, ktorý nadobúda konkrétne a oddelené hodnoty. Nie v každom čase nameriame informáciu.

Charakteristika: Jeho hodnota je definovaná v diskretných intervaloch a môže nadobúdať len obmedzený počet hodnôt.

Grafické znázornenie: Číslicový signál sa zvyčajne zobrazuje ako diskretný graf, kde sú hodnoty označené na určitých bodoch bez spojitých prechodov.

Digitálny signál:

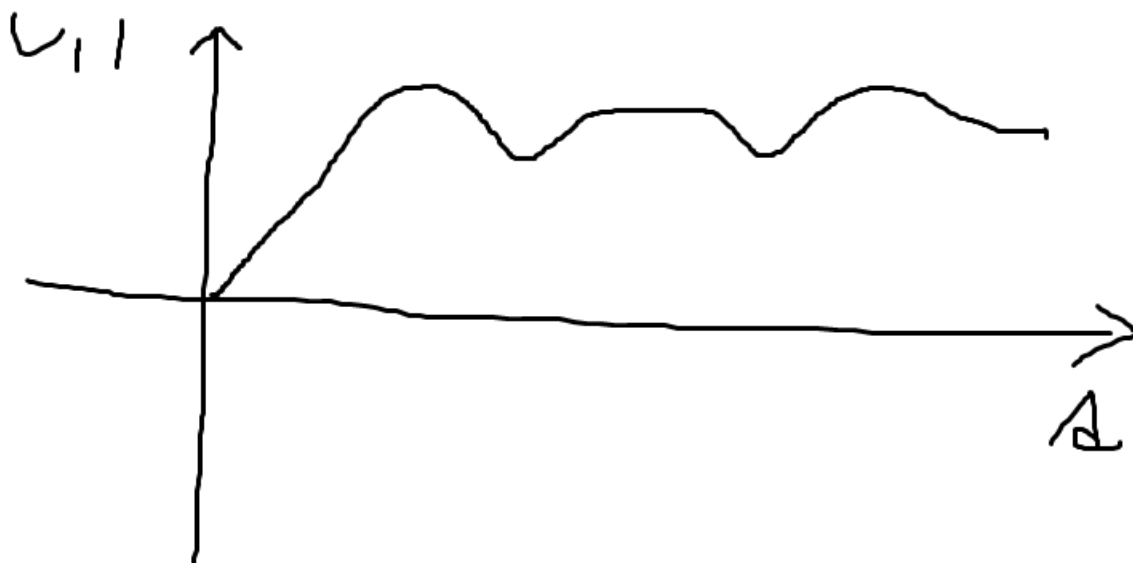
Reprezentácia: Digitálny signál je špeciálny typ číslicového signálu, ktorý je vyjadrený v binárnej podobe, kde sú použité iba dve hodnoty: 0 a 1. Má impulzy s rovnakou amplitúdou, ktoré sa líšia len šírkou medzery. V preklade: Keď hovoríme o impulzoch s rovnakou amplitúdou, znamená to, že všetky impulzy majú rovnakú intenzitu alebo silu. Rozdiel je len v šírke medzery medzi impulzmi.

Charakteristika: Je to najjednoduchší a najspoľahlivejší spôsob prenosu informácií, ktorý je odolný voči rušeniu a degradácii signálu.

Grafické znázornenie: Digitálny signál sa zobrazuje ako postupnosť binárnych čísel (0 a 1), ktoré sú diskontinuálne a oddelené.

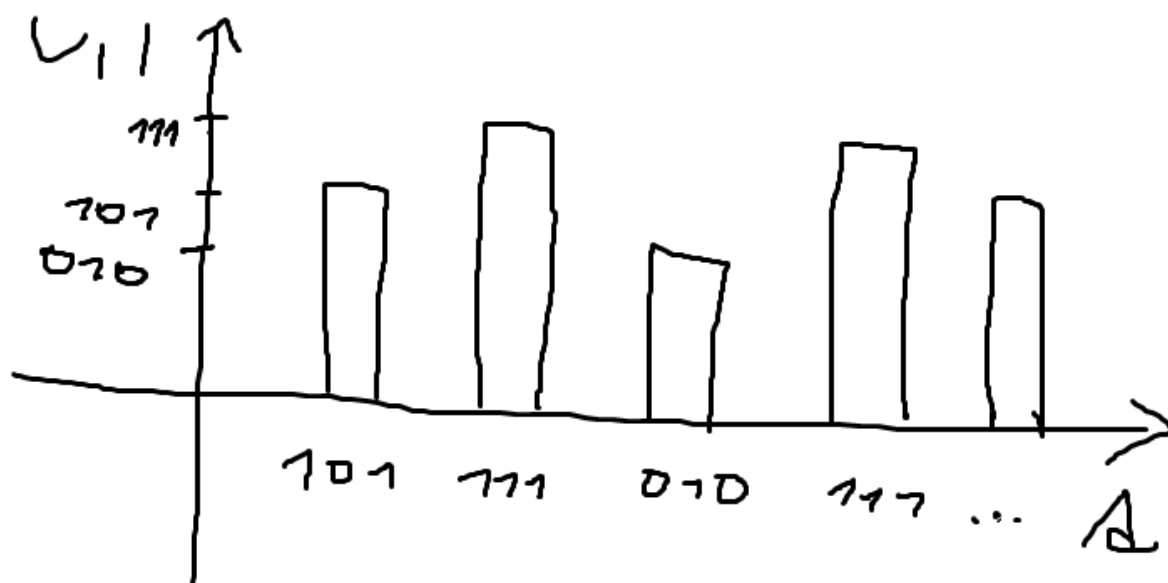
Analógový signál:

Na grafe by sme videli spojitú krivku, ktorá by sa postupne menila a mohla by nadobúdať ľubovoľné hodnoty v určenom rozsahu.



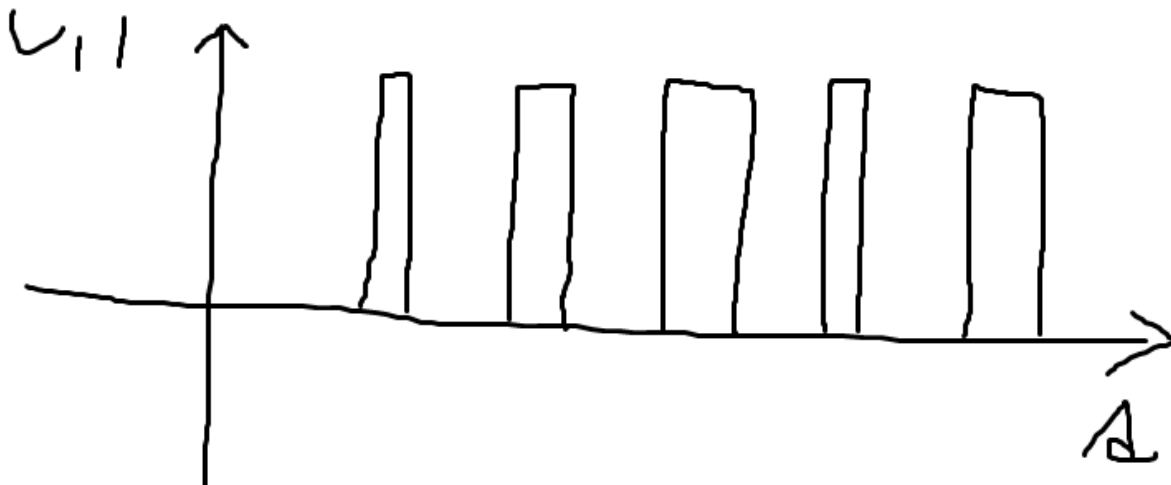
Číslicový signál:

Na grafe by sme videli sériu bodov alebo štvorcov, ktoré by reprezentovali diskkrétne hodnoty v určitých časových intervaloch. Medzi týmito bodmi by neboli žiadne spojité prechody.



Digitálny signál:

Na grafe by sme videli postupnosť binárnych čísel (0 a 1), ktoré by boli diskontinuálne a oddelené. Bodový graf by ukazoval, kedy je signál v stave "0" a kedy v stave "1".



a2 Objasnite význam digitalizácie informácií

Lepšia presnosť a kvalita: Pri digitálnych dátach môžeme presnejšie reprezentovať a uchovávať informácie, pretože digitálne dáta sú odolnejšie voči rušeniu, stratám a degradácii v porovnaní s analógovými dátami.

Jednoduchšie spracovanie: Digitálne údaje je ľahšie spracovať a manipulovať pomocou počítačov, pretože ich môžeme ľahko ukladať, kopírovať, prenášať, vyhľadávať a analyzovať.

Znížené náklady na skladovanie: Digitálne dáta môžeme efektívnejšie skladovať na elektronických médiách, čo znižuje potrebu fyzického skladovania a zvyšuje prístupnosť k informáciám.

Jednoduchšia distribúcia: Digitálne dáta môžeme jednoducho distribuovať cez internet, email, počítačové siete a iné elektronické prostriedky, čo umožňuje rýchlejšiu a efektívnejšiu výmenu informácií.

Viacnásobná využiteľnosť: Digitálne dáta môžeme ľahko konvertovať do rôznych formátov a používať vo viacerých aplikáciách, čo zvyšuje ich flexibilitu a využiteľnosť.

Vytváranie nových možností: Digitalizácia otvára cestu k vývoju nových technológií a inovácií v oblasti spracovania údajov, umelá inteligencia, rozpoznávanie obrazu a hlasu, a mnoho ďalších.

Možnosti analýzy a automatizácie: Digitálne dáta umožňujú ľahšie vykonávať analýzy, štatistiky a vyhľadávania, a poskytujú základ pre vývoj automatizovaných systémov a umelú inteligenciu.

a3 Popíšte časové priebehy na vstupoch a výstupoch časovačov TON a TOF

Časovač TON (časovač na oneskorenie):

Vstupný signál: Na vstupe časovača TON je typicky impulz, ktorý spúšťa časovač. Keď je vstupný signál aktivovaný (napríklad z 0 na 1), časovač začne odpočítavať.

Časový priebeh na vstupe: Na vstupe časovača TON bude impulz trvať po dobu, ktorá je rovná alebo dlhšia ako nastavený čas oneskorenia (časová hodnota T). Po uplynutí tohto času bude vstupný signál deaktivovaný.

Výstupný signál: Na výstupe časovača TON je vysielaný impulz po uplynutí nastaveného oneskorenia. Tento impulz sa zvyčajne aktivuje, keď uplynie čas T a deaktivuje sa, keď je časovač resetovaný alebo vstupný signál deaktivovaný.

Grafické znázornenie: Na grafe by sme videli nárastnú hranu impulzu na vstupe, ktorá trvá po dobu oneskorenia T . Výstupný impulz by bol zobrazený ako obdĺžnikový signál, ktorý sa objaví po uplynutí oneskorenia.

Časovač TOF (časovač na vypnutie):

Vstupný signál: Na vstupe časovača TOF je zvyčajne impulz alebo signál, ktorý spustí časovač. Keď je vstupný signál aktivovaný (napríklad z 0 na 1), časovač začne odpočítavať.

Časový priebeh na vstupe: Na vstupe časovača TOF bude impulz trvať po dobu, ktorá je rovná alebo kratšia ako nastavený čas vypnutia (časová hodnota T). Po uplynutí tohto času bude vstupný signál deaktivovaný.

Výstupný signál: Na výstupe časovača TOF je vysielaný impulz po uplynutí nastaveného času vypnutia. Tento impulz sa zvyčajne aktivuje po deaktivácii vstupného signálu a deaktivuje sa po uplynutí času T .

Grafické znázornenie: Na grafe by sme videli nárastnú hranu impulzu na vstupe, ktorá trvá po dobu aktívneho signálu. Výstupný impulz by bol zobrazený ako obdĺžnikový signál, ktorý sa objaví po uplynutí času vypnutia T .

a4 Popíšte základné kroky potrebné na digitalizáciu analógového signálu

Vzorkovanie (Sampling):

Prvým krokom je vzorkovanie analógového signálu. To znamená, že je získaná vzorka hodnôt signálu v pravidelných časových intervaloch. Tieto vzorky reprezentujú hodnoty analógového signálu v určitých diskretných okamihoch času.

Kvantizácia (Quantization):

Po vzorkovaní je každá vzorka hodnoty signálu zaokrúhlená alebo zaoblená na najbližšiu hodnotu v určitom diskretnom rozsahu. Tento proces sa nazýva kvantizácia. Výsledkom je diskretná množina možných hodnôt signálu.

Kódovanie (Encoding):

Každá kvantizovaná hodnota signálu je potom zakódovaná do digitálneho formátu. To znamená, že každá hodnota signálu je reprezentovaná sériou bitov, ktoré ju identifikujú a umožňujú jej uloženie a spracovanie v digitálnej podobe.

Ukladanie alebo prenos (Storage or Transmission):

Zakódované digitálne dáta môžu byť uložené na digitálnych médiách alebo prenesené cez digitálne komunikačné kanály. Digitálne dáta môžu byť ľahko spracované, ukladané a prenášané v porovnaní s analógovými dátami.

Reprodukcia (Reconstruction):

Pri prehrávaní alebo zobrazení digitálnych dát sa použije proces rekonštrukcie. To znamená, že digitálne dáta sú prevádzané späť na analógový signál pomocou algoritmov na dekódovanie a rekonštrukciu pôvodného signálu.

Filtrácia (Filtering) (voliteľné):

Vo väčšine prípadov môže byť potrebné aplikovať filtračné procesy na digitálny signál pred jeho rekonštrukciou. Tieto filtračné procesy môžu eliminovať šumy alebo artefakty, ktoré môžu byť prítomné v digitálnom signále.

Predstavme si, že máme analógový signál, napríklad zvukovú vlnu z mikrofónu. Tento signál je spojitý a môže mať rôzne hodnoty v priebehu času. Chceme ho digitalizovať, aby sme ho mohli uložiť alebo prenášať cez digitálny systém.

Vzorkovanie: Začneme tým, že budeme periodicky "pozerať" na našu zvukovú vlnu a získavať vzorky. Napríklad každých 0,01 sekundy zaznamenáme hodnotu zvuku.

Kvantizácia: Teraz, keď máme vzorky, musíme ich previesť na diskkrétne hodnoty. Povedzme, že máme kvantizačnú úroveň 8-bitov, čo znamená, že môžeme reprezentovať hodnoty od 0 po 255. Každý vzorek zvuku, napríklad 0,7 V, by sme zaokrúhlili na najbližšiu diskkrétne hodnotu, napríklad na 179.

Kódovanie: Tieto diskkrétne hodnoty reprezentujeme binárne. Napríklad hodnotu 179 by sme reprezentovali ako 10110011.

Prenos: Binárne kódy, ktoré sme získali, môžeme prenášať po počítačovej sieti alebo uložiť na digitálnom médiu.

Rekonštrukcia: Keď chceme znovu prehrať alebo zobrazíť tento zvuk, použijeme digitálno-analógový prevodník (DAC), ktorý premení naše binárne kódy späť na analógový signál. Tento signál by sme mohli prehrávať cez reproduktor a počuť pôvodný zvuk.

Takže, digitalizácia nám umožnila previesť pôvodný analógový zvuk na diskretný digitálny formát, ktorý je možné ľahko uložiť, prenášať a spracovávať pomocou digitálnych zariadení.

a5 Vysvetlite, aké finančné zdroje môže využívať výrobca PC a zaradte PC do majetku podniku

Vlastné finančné prostriedky:

Výrobca PC môže investovať do svojej činnosti pomocou vlastných finančných zdrojov. To zahŕňa zisk spoločnosti reinvestovaný späť do podniku alebo finančné prostriedky od majiteľov alebo akcionárov.

Pôžičky a úvery:

Spoločnosť môže získať finančné prostriedky prostredníctvom pôžičiek alebo úverov od bánk, finančných inštitúcií alebo iných zdrojov financovania. Tieto prostriedky môžu byť použité na investície do výroby, výskumu a vývoja, marketingu alebo ďalších podnikateľských aktivít.

Investície od investorov:

Výrobca PC môže získať finančné prostriedky prostredníctvom investícií od externých investorov, ako sú venture kapitálové spoločnosti, private equity fondy alebo anjeli investorí. Tieto investície môžu byť výmenou za podiel na spoločnosti alebo dohodnuté záruky v budúcnosti.

Ziskové a stratové účty:

Výrobca PC môže využívať zisky z predaja svojich výrobkov na financovanie svojich operácií a rastu. Zisky môžu byť reinvestované do podniku alebo vyplatené akcionárom ako dividendy.

Subvencie a dotácie:

Niektoré vlády alebo inštitúcie môžu poskytovať subvencie alebo dotácie pre spoločnosti, ktoré vyvíjajú technológie alebo produkty, ktoré majú strategický význam pre hospodárstvo alebo spoločnosť. Tieto prostriedky môžu byť použité na financovanie výskumu, vývoja alebo výroby.

PC sa považuje za majetok podniku, keď je zakúpený na účely podnikania a používaný v rámci podnikateľských operácií. Patrí do aktív spoločnosti a je zapísaný v jej účtovnej knihe. Hodnota počítača môže byť započítaná do bilancie spoločnosti ako súčasť jej majetku. Použitie počítača na podnikateľské účely môže byť tiež odpísané ako náklad v daňových pravidlách, čo môže znížiť daňové povinnosti spoločnosti.

b1 Popíšte PDU na L2

PDU (Protocol Data Unit) na úrovni 2 (L2) sa vzťahuje na dátovú jednotku používanú na druhej vrstve modelu OSI (Open Systems Interconnection). Druhá vrstva modelu OSI je známa ako Linková vrstva a je zodpovedná za prenos dát medzi zariadeniami v rovnakej lokálnej sieti. Tu je popis PDU na úrovni 2 a jej základné vlastnosti:

Rámec (Frame): Na úrovni 2 je PDU známa ako rámec (Frame). Rámec je jednotka dát, ktorá zahŕňa informácie potrebné na prenos dát medzi zariadeniami v rámci lokálnej siete.

Štruktúra rámca: Rámec zvyčajne obsahuje nasledujúce zložky:

Hlavička (Header): Obsahuje riadiace informácie potrebné pre správne doručenie rámca. To zahŕňa adresy zdroja a cieľa, typ rámca a kontrolnú sumu.

Dátové pole (Data Field): Obsahuje samotné užívateľské dáta alebo dátové jednotky prenášané medzi zariadeniami.

Päta (Footer): Môže obsahovať dodatočné informácie, ako napríklad kontrolné súčty alebo paritné bity, ktoré sa používajú na overenie integrity dát.

Adresovanie: Rámec obsahuje adresy zdroja a cieľa, ktoré umožňujú správne doručenie dátových jednotiek medzi zariadeniami v rámci siete. Tieto adresy sa zvyčajne používajú na identifikáciu sieťových rozhraní.

Typy rámcov: Existujú rôzne typy rámcov používaných na úrovni 2, vrátane Ethernetového rámca, Token Ring rámca a ďalších. Každý typ rámcov môže mať špecifické vlastnosti a štruktúru.

Funkcie linkovej vrstvy: Linková vrstva poskytuje niekoľko kľúčových funkcií vrátane detekcie a opravy chýb v prenose dát, riadenia prístupu k médiu, fragmentácie a reassembly dát, a riadenia toku dát.

b2 Popíšte spôsoby komunikácie half duplex a full duplex

Poloduplexná komunikácia (Half Duplex):

Charakteristika: V poloduplexnom režime môže zariadenie buď posielať, alebo prijímať dáta v určitom časovom okamihu, ale nie sú schopné robiť oboje súčasne.

Jednosmerná komunikácia: Komunikácia prebieha v jednom smere naraz, zatiaľ čo v opačnom smere je potrebné čakať na dokončenie prenosu alebo na signál, ktorý indikuje, že je miesto na odoslanie dát.

Použitie: Poloduplexná komunikácia sa často používa v jednoduchých systémoch, ako sú rádiové prenosy, walkie-talkies, a niektoré typy sietí Ethernet, ktoré vyžadujú delenie média.

Plnoduplexná komunikácia (Full Duplex):

Charakteristika: V plnoduplexnom režime môžu zariadenia súčasne posielať aj prijímať dáta. To znamená, že komunikácia môže prebiehať v oboch smeroch súčasne.

Dvojcestná komunikácia: Zariadenia môžu súčasne posielať a prijímať dáta, čo zvyšuje efektivitu komunikácie a znižuje oneskorenie.

Použitie: Plnoduplexná komunikácia sa často používa v moderných sieťových technológiách, ako sú siete Ethernet, Wi-Fi, Bluetooth, a ďalšie, ktoré vyžadujú rýchlu a spoľahlivú dvojcestnú komunikáciu.

Rozdiel:

Dvojcestná komunikácia: V plnoduplexnom režime môžu zariadenia komunikovať dvojcestne, zatiaľ čo v poloduplexnom režime komunikácia prebieha v jednom smere naraz.

Efektivita komunikácie: Plnoduplexná komunikácia poskytuje rýchlejšiu a spoľahlivejšiu výmenu dát v porovnaní s poloduplexnou komunikáciou.

Použitie a aplikácie: Plnoduplexná komunikácia sa často používa vo vysokorýchlostných sieťových technológiách a aplikáciách, ktoré vyžadujú rýchlu a efektívnu komunikáciu v oboch smeroch súčasne. Poloduplexná komunikácia sa používa v situáciách, kde dvojcestná komunikácia nie je nevyhnutná a je dostatočná jednosmerná komunikácia.

b3 Vysvetlite spôsob komunikácie na spoločnom médiu

CSMA/CD (Carrier Sense Multiple Access with Collision Detection):

Tento protokol je používaný v tradičných Ethernetových sieťach na riadenie prístupu k médiu.

Zariadenie, ktoré chce odoslať dáta, najprv počká na voľné médiá a potom odosiela svoje dáta. Avšak, predtým ako začne odosielať, monitoruje médiá, aby sa uistilo, že nie je žiadny iný prenos v tom istom čase.

Ak dve zariadenia začnú odosielať súčasne a ich signály sa prekrývajú, dochádza k kolízii. V takom prípade zariadenia zistia kolíziu a následne ju riešia opakovaním prenosu po určitej dobe náhodného čakania.

CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance):

Tento protokol sa používa v bezdrôtových sieťach, kde je detekcia kolízií ťažšia alebo nemožná.

Zariadenie monitoruje kanál a čaká na voľné médium pred odoslaním svojich dát. Avšak, namiesto odosielania dát bezprostredne po zistení voľného média, zariadenie ďalej čaká na určitú náhodnú dobu, aby sa minimalizovalo riziko kolízií.

Token Passing:

Tento protokol je používaný v sieťach typu Token Ring, kde je používaný token, ktorý sa pohybuje medzi zariadeniami.

Zariadenie môže odosielať dáta len vtedy, keď má fyzicky vlastný token. Po odoslaní dát token prechádza ďalej na ďalšie zariadenie.

Token slúži ako indikátor práva na odosielanie dát a eliminuje kolízie, ktoré by mohli nastať v prípade použitia CSMA/CD.

c1 Popíšte MAC tabuľku

MAC tabuľka (Media Access Control tabuľka) je tabuľka používaná v sieťových prepínačoch (switches) na uchovávanie informácií o tom, ktoré zariadenia sú pripojené na ktoré rozhrania switcha. Táto tabuľka sa používa na smerovanie paketov v rámci lokálnej siete (LAN).

Štruktúra MAC tabuľky:

MAC adresa zariadenia: Každý sieťový adaptér má svoju jedinečnú MAC adresu, ktorá slúži na identifikáciu zariadenia v sieti. V MAC tabuľke sa ukladajú tieto MAC adresy.

Port/rozhranie switcha: Pre každú MAC adresu v tabuľke sa ukladá informácia o tom, na ktorom portu alebo rozhraní switcha je dané zariadenie pripojené.

Časové razítko (timestamp): V niektorých prípadoch sa do MAC tabuľky ukladá aj časové razítko, ktoré indikuje, kedy bola poslednýkrát zaznamenaná aktivita od daného zariadenia. To umožňuje switchu aktualizovať informácie v tabuľke a odstrániť zariadenia, ktoré už nie sú aktívne v sieti.

Funkcia MAC tabuľky:

Riadenie preposielania paketov: Keď switch dostane paket, pozrie sa na cieľovú MAC adresu v pakete. Potom sa pozrie do svojej MAC tabuľky, aby sa zistilo, na ktorom porte sa nachádza zariadenie s cieľovou MAC adresou. Paket sa potom prepošle iba na daný port, na ktorom sa nachádza cieľové zariadenie, čím sa znižuje zahľtenie siete a zvyšuje jej efektívnosť.

Učenie sa a aktualizácia tabuľky: Switchi neustále monitorujú komunikáciu v sieti a aktualizujú svoju MAC tabuľku podľa potreby. Keď switch zistí novú MAC adresu na nejakom porte (napríklad, keď zariadenie pošle paket), pridá túto MAC adresu do svojej tabuľky spolu s informáciou o porte, na ktorom bolo zariadenie zaznamenané.

Aging (starnutie): Niektoré záznamy v MAC tabuľke môžu byť automaticky odstránené po určitom časovom období, ktoré sa nazýva aging time. To umožňuje odstrániť záznamy pre zariadenia, ktoré už nie sú aktívne v sieti a uvoľniť pamäť v tabuľke pre nové zariadenia.

c2 Popíšte manažovanie switchu na L2

Konfigurácia portov:

Nastavenie VLAN (Virtual LAN): Administrátor môže konfigurovať VLAN na jednotlivých portoch switchu, čo umožňuje segmentáciu siete a izoláciu prevádzky medzi rôznymi skupinami zariadení.

Duplex a rýchlosť portov: Administrátor môže nastaviť rýchlosť a duplex na jednotlivých portoch podľa potreby siete.

Nastavenie bezpečnosti portov: Konfigurácia bezpečnosti portov môže zahŕňať obmedzenie počtu povolených MAC adries na porte, aktiváciu port security, alebo obmedzenie prístupu na základe MAC adries alebo VLAN.

Správa VLAN:

Vytváranie VLAN: Administrátor môže vytvárať a konfigurovať VLAN podľa potreby siete.

Riadenie toku dát medzi VLAN: Konfigurácia prístupových a trunkových portov umožňuje riadenie toku dát medzi VLAN, čo umožňuje alebo obmedzuje komunikáciu medzi rôznymi VLAN v sieti.

Monitorovanie a diagnostika:

Monitorovanie prevádzky portov: Administrátor môže sledovať prevádzku na jednotlivých portoch switchu, aby identifikoval prípadné problémy alebo zahŕtenie siete.

Diagnostika spojenia: Switch môže poskytovať informácie o fyzickom stave portov a pripojených zariadeniach, vrátane detekcie chýb, zlých káblov, alebo neaktívnych portov.

Zálohovanie a obnovenie konfigurácie:

Zálohovanie konfigurácie: Administrátor môže pravidelne zálohovať konfiguráciu switchu, aby sa zabezpečilo, že sa dá obnoviť v prípade potreby.

Obnovenie konfigurácie: V prípade výpadku alebo straty konfigurácie môže byť konfigurácia obnovená zo zálohy, aby sa minimalizovalo prerušenie prevádzky.

3. Elektronika – Lineárne a nelineárne elektronické prvky - Samo

1. Porovnajzte rozdiel medzi lineárnymi a nelineárnymi elektronickými prvkami
2. Vysvetlite princíp merania odporu priamou a nepriamou metódou
3. Vysvetlite rozdiel medzi ideálnymi a reálnymi elektronickými prvkami
4. Porovnajzte správanie sa prvkov R, L a C v obvodoch jednosmerného a striedavého prúdu
5. Určte hodnotu predradného odporu LED diódy s prahovým napätím 2V, ktorou má pretekať prúd 20 mA, ak je napájacie napätie 12 V

b)

1. Popíšte proces nábehu pri zapnutí zariadenia switch - LED diódy
2. Definujte úlohy fyzickej vrstvy modelu OSI
- 3.
4. Popíšte TP káble

c)

1. Načrtnite štruktúru optických káblov
2. Zaradte PC do majetku podniku a popíšte jeho obstaranie, ocenenie ako dlhodobého majetku

a1 Porovnajzte rozdiel medzi lineárnymi a nelineárnymi elektronickými prvkami

Lineárne prvky:

Lineárne prvky majú konštantný pomer medzi výstupnou a vstupnou veličinou, ktorý sa nepodmieňuje veľkosťou vstupného signálu. Napríklad, ak na nich zvýšite napätie o určitú hodnotu, výstup sa zvýši v rovnakom pomere.

Príklady zahŕňajú rezistory a ideálne operačné zosilňovače (v určitom rozsahu).

Nelineárne prvky:

Nelineárne prvky majú variabilný pomer medzi výstupnou a vstupnou veličinou, ktorý sa môže meniť v závislosti od veľkosti vstupu. To znamená, že ich chovanie nie je lineárne.

Ich výstup sa môže meniť v rôznych pomeroch vzhľadom na zmenu vstupu. Príklady zahŕňajú diódy, tranzistory, a mnohé polovodičové prvky.

V praxi sú lineárne prvky často používané na realizáciu statických funkcií, ako sú zosilňovače s konštantným zosilnením. Naopak, nelineárne prvky sa často používajú na realizáciu dynamických funkcií, ako sú spínanie, usmerňovanie a podobne.

a2 Vysvetlite princíp merania odporu priamou a nepriamou metódou

Priama metóda merania odporu:

Princíp: Pri priamej metóde sa odpor meria priamo pomocou odporového meradla, ako je multimeter. Multimeter je zariadenie, ktoré môže merať rôzne elektrické vlastnosti, vrátane odporu. Priame meranie zahŕňa pripojenie zariadenia na meraný odpor a priame zobrazenie jeho hodnoty na displeji.

Postup:

Pripojte merací prístroj (multimeter) k obvodu alebo prvku, ktorého odpor chcete zmerať.

Zvoľte režim merania odporu na multimeteri.

Zmerajte hodnotu odporu.

Výhody:

Jednoduché a priame.

Rýchle a presné meranie.

Nepriama metóda merania odporu:

Princíp: Pri nepriamej metóde sa odpor určuje pomocou známych vlastností obvodu, v ktorom sa daný odpor nachádza. To môže zahŕňať meranie napätia a prúdu v obvode a použitie Ohmového zákona ($R = V/I$) na výpočet hodnoty odporu.

Postup:

Merajte napätie (V) na odporovom prvku.

Merajte prúd (I), ktorý cez neho prechádza.

Použite Ohmov zákon na výpočet odporu ($R = V/I$).

Výhody:

Nepriame metódy môžu byť využité aj v prípadoch, keď nie je možné odpor priamo zmerať.

Umožňuje meranie odporu v zložitejších obvodoch.

Porovnanie:

Priama metóda je priamočiarejšia a rýchlejšia, ale nie vždy je možné odpor priamo zmerať, najmä v prípadoch, keď je súčasťou zložitého obvodu. V takýchto prípadoch je vhodné použiť nepriame metódy merania odporu.

a3 Vysvetlite rozdiel medzi ideálnymi a reálnymi elektronickými prvkami

Ideálne elektronické prvky:

Princíp: Ideálne prvky sú modelované na základe jednoduchých ideálnych predpokladov, ktoré zjednodušujú analýzu a návrh obvodov. Sú to teoretické koncepty, ktoré majú určité vlastnosti, ktoré sa v praxi často nepodarí dosiahnuť.

Vlastnosti:

Nemajú žiadny vnútorný odpor alebo straty energie.

Ich parametre sú konštantné a nezávislé na teplotnej alebo iných podmienkach.

Napríklad ideálny rezistor má presne danú hodnotu odporu bez akejkoľvek variácie.

Použitie:

Používajú sa najmä v teoretických analýzach a modeloch obvodov, aby sa zjednodušili výpočty a pochopenie princípov fungovania.

Reálne elektronické prvky:

Princíp: Reálne elektronické prvky sú skutočné komponenty, ktoré sa používajú v praktických aplikáciách. Na rozdiel od ideálnych prvkov majú reálne prvky nedokonalosti a ich správanie sa môže líšiť v závislosti od rôznych faktorov.

Vlastnosti:

Majú vnútorný odpor a straty energie.

Ich parametre môžu byť ovplyvnené teplotou, frekvenciou, napätím a ďalšími faktormi.

Napríklad reálny rezistor môže mať malý vnútorný odpor a jeho hodnota sa môže mierne meniť v závislosti od teploty a ďalších faktorov.

Použitie:

Používajú sa vo všetkých praktických aplikáciách elektroniky, pretože zrkadlia skutočné podmienky a obmedzenia, ktoré ovplyvňujú výkon a správanie elektronických systémov.

Celkovo ideálne prvky poskytujú jednoduchší model na analýzu obvodov, zatiaľ čo reálne prvky zohľadňujú skutočné podmienky a obmedzenia, ktoré sa vyskytujú v praktických aplikáciách.

a4 Porovnajte správanie sa prvkov R, L a C v obvodoch jednosmerného a striedavého prúdu

Jednosmerný prúd:

Rezistor (R): V obvode s jednosmerným prúdom sa správa rezistoru nezmení. Rezistor jednoducho obmedzuje prúd, ktorý cez neho prechádza, a konvertuje elektrickú energiu na teplo podľa Ohmovho zákona ($U = IR$).

Induktor (L): Pri jednosmernom prúde sa induktor správa ako otvorený obvod. To znamená, že sa správa ako veľmi veľký odpor pre jednosmerný prúd, takže takmer žiadny prúd nekrúti. Jeho jednosmerný odpor je teoreticky nekonečný.

Kondenzátor (C): Pri jednosmernom prúde sa kondenzátor správa ako krátky obvod. To znamená, že v okamihu zapnutia kondenzátoru sa všetka energia akumulovaná v elektrickom poli na ňom vyčerpá. Jeho jednosmerný odpor je teoreticky nulový.

Striedavý prúd:

Rezistor (R): Správanie rezistoru v obvode so striedavým prúdom je rovnaké ako pri jednosmernom prúde. Rezistor jednoducho obmedzuje prúd podľa Ohmového zákona a mení elektrickú energiu na teplo.

Induktor (L): Pri striedavom prúde sa induktor správa ako zložitý prvok, ktorý vytvára magnetické pole, ktoré sa mení v čase. V dôsledku toho vzniká indukované napätie, ktoré spôsobuje v obvode proti prúdu. Tento efekt spôsobuje zdĺhavý nárast prúdu cez induktor.

Kondenzátor (C): Pri striedavom prúde sa kondenzátor správa ako zložitý prvok, ktorý uchováva náboj a vytvára elektrické pole. Pri meniacom sa napätí na kondenzátore sa náboj pohybuje dovnútra a von z kondenzátora v závislosti od polarít napätia, čo vedie k prúdu.

V striedavých obvodoch teda induktor a kondenzátor majú dynamické správanie, ktoré závisí od frekvencie striedavého prúdu, zatiaľ čo rezistor sa správa staticky bez ohľadu na frekvenciu.

a5 Určte hodnotu predradného odporu LED diódy s prahovým napätím 2V, ktorou má pretekať prúd 20 mA, ak je napájacie napätie 12 V

Ohmov zákon hovorí, že napätie na rezistore (predradnom odpore) je rovnaké ako rozdiel medzi napájacím napätím a napätím na LED dióde (prahové napätie LED). Pri zadanom prúde cez LED diódu a požadovanom prahovom napätí môžeme určiť hodnotu predradného odporu pomocou vzťahu:

$$R = \frac{V_{supply} - V_{LED}}{I}$$

Kde:

- V_{supply} je napájací zdroj (12 V),
- V_{LED} je prahové napätie LED diódy (2 V),
- I je požadovaný prúd LED diódy (20 mA alebo 0.020 A).

Teraz dosadíme známe hodnoty do vzťahu:

$$R = \frac{12V - 2V}{0.020A}$$

$$R = \frac{10V}{0.020A}$$

$$R = 500\Omega$$

Takže hodnota predradného odporu LED diódy by mala byť približne 500 Ohmov. Je dôležité vybrať najbližšiu dostupnú štandardnú hodnotu odporu, ktorá je rovnaká alebo väčšia ako vypočítaná hodnota.

b1 Popíšte proces nábehu pri zapnutí zariadenia switch - LED diódy

Proces nábehu (alebo spínacieho času) pri zariadení s prepínačom a LED diódou sa týka času potrebného na dosiahnutie plného svietenia LED diódy po zapnutí prepínača. Tento proces závisí od rôznych faktorov, ako sú kapacita kondenzátora (ak sa používa), vnútorný odpor zdroja napájania a samotného prepínača, ako aj časové konštanty obvodu.

Zapnutie prepínača:

Keď používateľ zapne prepínač, vytvorí sa spojenie v obvode a elektrický prúd začne tiecť cez LED diódu.

Nábeh napätia:

V okamihu zapnutia prepínača je počiatočné napätie na LED dióde nízke. Postupne sa zvyšuje v dôsledku nabíjania kondenzátora (ak je prítomný) a prechodu prúdu cez diódu.

Nárast prúdu:

S nárastom napätia na LED dióde sa zvyšuje aj prúd, ktorý cez ňu prechádza. LED dióda postupne dosahuje svoju nominálnu pracovnú hodnotu a začne svietiť.

Dosiahnutie plného svietenia:

Po určitom čase potrebnom na nabitie kondenzátora a dosiahnutie plného prúdu pre LED diódu, dosiahne LED dióda svoje maximálne svietenie.

Stabilizácia:

Keď sa LED dióda dostane do stabilného stavu, prúd a napätie v obvode sa ustália na určitých hodnotách. Čas, ktorý tento proces trvá, môže byť veľmi krátky, pokiaľ je zdroj napájania stabilný a reakcia obvodu rýchla.

b2 Definujte úlohy fyzickej vrstvy modelu OSI

Fyzická vrstva modelu OSI (Open Systems Interconnection) je prvá z vrstiev tohto modelu. Je zodpovedná za fyzickú prenosovú a prijímaciu časť dát medzi zariadeniami a riadi prístup k fyzickému médium, ako sú koaxiálne káble, optické vlákna alebo bezdrôtové prenosové médium. Tu sú hlavné úlohy fyzickej vrstvy:

Prepojenie fyzického média: Fyzická vrstva umožňuje prenos dát cez rôzne typy fyzických médií, ako sú káble, vlákna alebo bezdrôtové prenosové médiá. Zabezpečuje, že dáta sú vhodne zakódované pre dané médium a prenášajú sa cezň správne.

Synchronizácia dátových rámcov: Fyzická vrstva zabezpečuje, že dáta sú prenášané v správnom čase a že odosielateľ a prijímateľ sú synchronizovaní. To zahŕňa koordináciu bitov na fyzickom médiu tak, aby boli dáta presne prenesené a následne interpretované správne.

Fyzická topológia siete: Fyzická vrstva definuje fyzickú topológiu siete, teda spôsob, ako sú zariadenia pripojené a ako sú spojené s prenosovými médiami. Môže ísť o hviezdicovú, kruhovú, zbernícovú alebo inú topológiu, ktorá ovplyvňuje spôsob, ako sa dáta fyzicky prenášajú.

Kódovanie a modulácia signálu: Fyzická vrstva zahŕňa procesy kódovania a modulácie signálu, ktoré umožňujú prevádzať digitálne dáta na analógový signál pre prenos cez fyzické médium a následné dekódovanie a demoduláciu na prijímateľskej strane.

Detekcia a oprava chýb: V niektorých prípadoch môže fyzická vrstva zahŕňať metódy detekcie a opravy chýb v prenesených dátach, napríklad prostredníctvom redundancie bitov alebo kontrolných súčtov.

b3 Popíšte TP káble

TP (twisted pair) káble sú druhom káblového média používaného na prenos dát v počítačových sieťach a telekomunikáciách. Tieto káble sa skladajú z párných vodičov, ktoré sú do seba zabalené a vzájomne zvinuté, aby sa minimalizovala interferencia a šírenie elektromagnetických rušení.

Dvojitý zvinutý vodič: Každý pár vodičov v TP kábli sa skladá z dvoch drôtov, ktoré sú zvinuté do seba. Toto zvinutie pomáha minimalizovať elektromagnetickú interferenciu (EMI) a krosstalk (prechod signálu medzi pármami), čo zvyšuje spoľahlivosť prenosu dát.

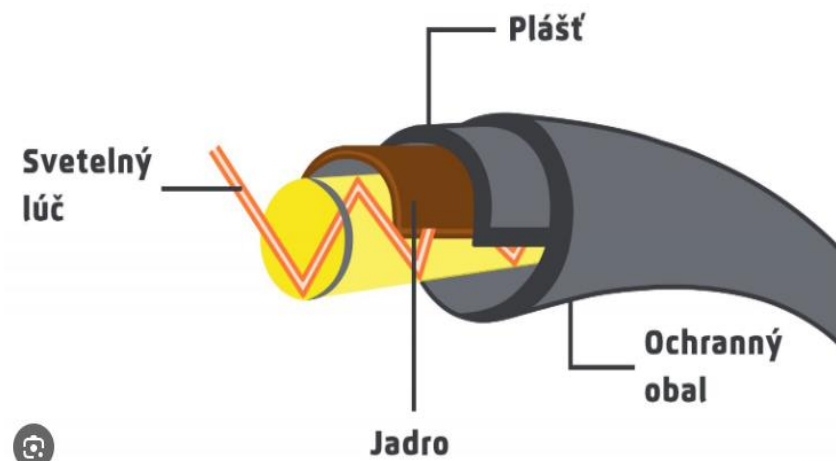
Kategórie káblov: TP káble sú kategorizované podľa ich schopnosti prenášať dáta pri určitej frekvencii. Najbežnejšie kategórie sú Cat5e, Cat6 a Cat6a. Novšie kategórie umožňujú vyššie rýchlosti prenosu dát a lepšiu odolnosť proti interferencii.

Konektory: Pre pripojenie TP káblov sa bežne používajú konektory typu RJ-45. Tieto konektory majú 8 kontaktov, ktoré sú používané na prenos dát a napájanie v prípade potreby (napríklad pre IP telefóny alebo IP kamery).

Použitie: TP káble sú široko používané v počítačových sieťach pre Ethernetové pripojenia, ako aj v telekomunikáciách pre telefónne linky a iné typy prenosu dát. Sú flexibilné, lacné a jednoduché na inštaláciu.

Rôzne varianty: Okrem klasických neuzemnených TP káblov existujú aj uzemnené varianty (STP - shielded twisted pair), ktoré majú okolo vodičov dodatočnú ochrannú vrstvu na ochranu pred elektromagnetickými rušeniami.

c1 Načrtnite štruktúru optických káblov



c2 Zaradíte PC do majetku podniku a popíšte jeho obstaranie, ocenenie ako dlhodobého majetku

Identifikácia potreby a obstaranie: Najprv manažment podniku identifikuje potrebu počítačov pre rôzne oddelenia alebo zamestnancov. Potom sa vykonáva proces obstarania, ktorý môže zahŕňať vyhodnotenie požiadaviek, výber vhodných dodávateľov, vyjednávanie cien a podpísanie zmlúv.

Nákup a dodanie: Po schválení obstarávacieho procesu a nákupu sú počítače dodané do podniku a nainštalované na pracoviskách.

Inventarizácia a evidencia majetku: Po dodaní sa počítače zaradia do inventára podniku. Každý počítač je evidovaný ako dlhodobý majetok a priradený unikátnemu identifikačnému číslu alebo kódu.

Hodnotenie a ocenenie: Počítače sú ocenené ako dlhodobý majetok na základe ich počiatočnej nákupnej ceny. Toto ocenenie zahŕňa nielen cenu počítača, ale aj náklady na jeho inštaláciu, konfiguráciu a prípadné ďalšie potrebné vybavenie (napríklad monitor, klávesnica, myš).

Zaradenie do účtovníctva: Počítače ako dlhodobý majetok sú zaradené do účtovníctva podniku ako dlhodobý majetok na príslušných účtovných kategóriách. Ich hodnota sa amortizuje v čase a odráža sa v účtovnej bilancii podniku.

Správa a údržba: Po zaradení do majetku sa počítače pravidelne spravujú a udržiavajú, aby sa zabezpečila ich dlhá životnosť a efektívna prevádzka.

4. Elektronika – Základné zákony elektrotechniky a riešenie elektrických obvodov - Samo

a)

1. Definujte Ohmov zákon a Kirchhoffove zákony
2. Popíšte rozdiel medzi elektrickým prúdom, napätím a odporom
3. Vysvetlite pojmy: slučka elektrického obvodu, elektrický uzol a skrat
4. Popíšte metódu merania odporov Volt-Ampérovou metódou
5. Porovnajte rozdiel medzi vnútorným odporom voltmetra a ampérmetra

b)

1. Porovnajte výpočet zaťaženého a nezaťaženého napäťového deliča aplikovaním Ohmovho a Kirchhoffových zákonov
2. Definujte pojem autorské právo v kontexte software-u
3. Rozdeľte zariadenia podľa vrstvy komunikačného modelu, na ktorej pracuje

c)

1. Načrtnite typy topológie sietí
2. Charakterizujte daňovú sústavu, druhy dani a vysvetlite vzťah medzi štátnym rozpočtom a daňami

a1 Definujte Ohmov zákon a Kirchhoffove zákony

Ohmov zákon:

Definícia: Ohmov zákon je základný princíp v elektrickej teórii, ktorý popisuje vzťah medzi napätím, prúdom a odporom v elektrickom obvode.

Matematický vzťah: Ohmov zákon sa vyjadruje vzťahom

$$U = I \times R$$

$U = I \times R$, kde:

$U = U$ je napätie v obvode vo voltoch (V),

$I = I$ je prúd v obvode v ampéroch (A),

$R = R$ je odpor v obvode v ohmoch (Ω).

Význam: Ohmov zákon hovorí, že napätie medzi dvoma bodmi v elektrickom obvode je priamo úmerné prúdu, ktorý cez ne prechádza, a odporu obvodu.

Kirchhoffove zákony:

Kirchhoffove zákony sú dva základné zákony používané na analýzu a výpočet prúdov a napätí v elektrických obvodoch.

Kirchhoffov prvý zákon (KZP), známy aj ako zákon o prúdoch:

Definícia: Suma prúdov vstupujúcich do uzlu v uzavretom obvode je rovná sume prúdov vystupujúcich z tohto uzlu.

Význam: Kirchhoffov prvý zákon vyjadruje zachovanie náboja v uzavretom obvode a umožňuje analýzu prúdových rozvetvení v obvode.

Kirchhoffov druhý zákon (KZD), známy aj ako zákon o napätiach:

Definícia: V uzavretom smyčke obvodu je algebrická suma napätí rovná nule.

Význam: Kirchhoffov druhý zákon umožňuje analýzu napäťových dropov v obvode a vyjadruje zachovanie energie v uzavretých smyčkách.

a2 Popíšte rozdiel medzi elektrickým prúdom, napätím a odporom

Elektrický prúd (I):

Definícia: Elektrický prúd je tok nábojových častíc, ktoré sa pohybujú cez vodič v elektrickom obvode. Vyjadruje množstvo elektrických nábojov, ktoré prechádzajú za jednotku času cez daný bod obvodu.

Jednotka: Ampér (A).

Symbol: I

Vlastnosti: Prúd môže byť striedavý alebo jednosmerný, a môže byť konštantný alebo meniaci sa v čase.

Elektrické napätie (V):

Definícia: Elektrické napätie je rozdiel potenciálov medzi dvoma bodmi v elektrickom obvode. Vyjadruje množstvo práce, ktorú musíme vykonať na prenesenie jednotkového náboja z jedného bodu do druhého.

Jednotka: Volt (V).

Symbol: V

Vlastnosti: Napätie môže byť konštantné alebo meniace sa v čase (striedavé napätie).

Elektrický odpor (R):

Definícia: Elektrický odpor je schopnosť látky alebo prvku obmedzovať tok elektrického prúdu. Vyjadruje sa ako pomer medzi napätím a prúdom v obvode.

Jednotka: Ohm (Ω).

Symbol: R

Vlastnosti: Odpor závisí od materiálu, z ktorého je vodič vyrobený, jeho dĺžky, prierezu a teploty. Materiály s vyššou elektrickou vodivosťou majú nižší odpor.

Rozdiely:

Elektrický prúd je tok náboja v obvode, napätie je rozdiel potenciálu medzi dvoma bodmi v obvode a odpor je schopnosť obmedzovať tok prúdu v obvode.

Prúd a napätie sú aktívne veličiny, ktoré sa merajú v reálnom čase, zatiaľ čo odpor je pasívna vlastnosť materiálu alebo prvku v obvode.

Prúd a napätie sú priamo závislé od seba prostredníctvom Ohmového zákona ($V=I \times R$) kde R predstavuje odpor v obvode.

a3 Vysvetlite pojmy: slučka elektrického obvodu, elektrický uzol a skrat

Slučka elektrického obvodu:

Slučka v elektrickom obvode je uzavretá cesta, po ktorej môže elektrický prúd prejsť. To znamená, že prúd môže putovať po obvode, začínať a končiť v tom istom bode. Slučka je dôležitý koncept pri analýze elektrických obvodov, pretože umožňuje porozumieť tomu, ako prúd prechádza rôznymi časťami obvodu.

Elektrický uzol:

Elektrický uzol je bod v elektrickom obvode, kde sa stretávajú tri alebo viac vodičov. V uzle sa prúd rozdeľuje medzi väčší počet vetiev obvodu. Zákon zachovania náboja hovorí, že súčet prúdov vstupujúcich do uzlu musí byť rovný súčtu prúdov vystupujúcich z uzlu. Elektrické uzly sú základným stavebným prvkom pri analýze a návrhu elektrických obvodov.

Skrat:

Skrat je nežiaduci udalosť v elektrickom obvode, kedy dochádza k priamemu spojeniu medzi dvoma vodičmi s rôznymi potenciálmi alebo k priamemu spojeniu medzi fázou a zemou. Toto spojenie spôsobuje výrazný prúd, ktorý môže spôsobiť požiar, poškodenie zariadení alebo úrazy. Skraty sú nebezpečné a je dôležité mať ochranné zariadenia, ako sú poistky alebo ochranné prepínače, ktoré chránia obvod pred preťažením a skratmi.

a4 Popíšte metódu merania odporov Volt-Ampérovou metódou

Metóda merania odporov pomocou Volt-Ampérovej metódy, často nazývaná aj ako metóda dvojčlenná, je jednoduchá a efektívna technika na určenie odporu elektrického prvku. Táto metóda využíva Ohmov zákon ($U = I \cdot R$), kde U je napätie, I je prúd a R je odpor.

Postup merania je nasledovný:

Pripojenie obvodu:

Najprv sa pripojí zdroj napätia (napríklad batéria) k meranému odporu. Tento obvod bude tvorený sériovým spojením zdroja napätia, meraného odporu a ampérmetra.

Nastavenie napätia:

Zdroj napätia sa nastaví na známu hodnotu napätia. Toto napätie sa obvykle volí tak, aby bolo jednoduché merať prúd, ale zároveň aby nedošlo k poškodeniu odporu.

Meranie prúdu:

Prúd, ktorý prechádza obvodom, sa meria pomocou ampérmetra, ktorý je zapojený paralelne s meraným odporovým prvkom. Ampérmetra by mal mať čo najnižší vlastný odpor, aby neovplyvnil meranie.

Výpočet odporu:

Po zmere prúdu a znalosti napätia sa odpor meraného prvku vypočíta pomocou Ohmového zákona, teda $R = U / I$, kde U je napätie a I je prúd.

Táto metóda je obvykle presná a relatívne jednoduchá na vykonanie, najmä ak máte k dispozícii meracie prístroje ako napäťový zdroj, ampérmetra a prípadne multimeter na presné meranie napätia a prúdu. Je vhodná na meranie odporov v širokom rozsahu hodnôt, pokiaľ sú dodržané bezpečnostné a výkonnostné limity používaných komponentov.

a5 Porovnajete rozdiel medzi vnútorným odporom voltmetra a ampérmetra

Definícia:

Vnútorný odpor voltmetra: Je to odpor, ktorý má voltmeter a ktorý obmedzuje prúd, ktorý môže prejsť cez voltmetre. To znamená, že voltmeter odporuje prúdu, čím minimalizuje jeho vplyv na obvod, ktorý meria.

Vnútorný odpor ampérmetra: Je to odpor, ktorý má ampérmetra a ktorý obmedzuje napätie, ktoré spôsobuje prúd, ktorý prechádza cez ampérmetra. Tým minimalizuje vplyv napätia na obvod, ktorý meria.

Umiestnenie v obvode:

Vnútorný odpor voltmetra: Je pripojený paralelne k meranému zariadeniu, čo znamená, že väčšina prúdu ide cez merané zariadenie a iba malý prúd ide cez voltmeter.

Vnútorný odpor ampérmetra: Je pripojený sériovo k meranému zariadeniu, takže celý prúd, ktorý prechádza meraným zariadením, ide cez ampérmetra.

Hodnoty odporu:

Vnútorý odpor voltmetra: Je vysoký, aby minimalizoval prúd, ktorý prechádza cez voltmeter a minimalizoval vplyv na meraný obvod.

Vnútorý odpor ampérmetra: Je nízky, aby minimalizoval napätie spôsobené prúdom, ktorý prechádza cez ampérmetra a aby sa minimalizovala strata napätia.

Meranie:

Vnútorý odpor voltmetra: Nie je priamo merateľný, ale môže byť odvodený z charakteristík voltmetra a znalosti vnútorných obvodov.

Vnútorý odpor ampérmetra: Môže byť meraný priamo pomocou externého zdroja napätia a rezistora a aplikovaním Ohmového zákona.

b1 Porovnajte výpočet zaťaženého a nezaťaženého napäťového deliča aplikovaním Ohmovho a Kirchoffových zákonov

Nezaťažený napäťový delič:

Popis: Nezaťažený napäťový delič je taký, kde k výstupu nie je pripojený žiadny externý odpor.

Výpočetný vzorec: Napätie na výstupe nezaťaženého deliča (U_v) je jednoducho dané ako podiel vstupného napätia (U_{in}) a pomeru odporov deliča (R_1 a R_2). Vzorec je: $U_v = U_{in} \times R_2 / (R_1 + R_2)$

Význam: Tento výpočet poskytuje napätie na výstupe deliča za ideálnych podmienok, kedy prúd tečie len cez rezistory deliča.

Zaťažený napäťový delič:

Popis: Zaťažený napäťový delič je taký, kde k výstupu je pripojený externý záťažový odpor (R_{load}).

Výpočetný vzorec: Napätie na výstupe zaťaženého deliča (U_v) sa vypočíta pomocou vzťahu napätia deliča bez záťaže (U_{v_nezat}) a vnútorného odporu záťaže (R_{load}), ako aj pomeru odporov deliča (R_1 a R_2). Vzorec je:

$$U_v = U_{v_nezat} \times R_{load} / (R_{load} + R_2)$$

Význam: Tento výpočet zohľadňuje vplyv záťažového odporu, čo môže ovplyvniť výstupné napätie deliča v dôsledku vnútorného odporu záťaže.

Celkovo vzorec pre zaťažený delič je odvodený z nezaťaženého deliča tak, že sa k nemu pridá vplyv záťažového odporu, čo je kľúčový rozdiel medzi týmito dvoma metódami výpočtu.

b2 Definujte pojem autorské právo v kontexte software-u

Autorské právo v kontexte softwaru sa týka právnych zásad a ochranných opatrení, ktoré chránia originálnu tvorivú prácu vytvorenú v rámci softvérového produktu. Tieto práva poskytujú autorovi alebo držiteľovi práv na softvér právnu ochranu proti neoprávnenému používaniu, kopírovaniu, distribúcii alebo modifikácii jeho diela. Tu sú kľúčové prvky autorského práva v kontexte softwaru:

Originálna tvorivá práca: Softvér je považovaný za originálnu tvorivú prácu, pretože vyžaduje kreatívne úsilie a ochranu pre vývoj a implementáciu nových myšlienok a konceptov.

Ochrana formy, nie obsahu: Autorské právo chráni konkrétne spôsoby, akými je softvér vyjadrený, nie však jeho funkčnosť alebo myšlienky. To znamená, že kód softvéru je chránený, ale nie nápady, ktoré môžu byť v kóde obsiahnuté.

Práva autora: Autor softvéru má právo na kontrolu kopírovania, distribúcie, verejného vystavenia, vykonávania a modifikácie jeho diela.

Licenčné ujednanie: Užívatelia softvéru sú obvykle oprávnení používať softvér v súlade s podmienkami uvedenými v licenčnej zmluve. Licenčná zmluva môže obmedzovať určité aspekty užívania softvéru, ako je napríklad možnosť zmien alebo redistribúcie.

Dĺžka ochrany: Autorské práva na softvér obvykle platia počas života autora plus určitý počet rokov (väčšinou 50 alebo 70 rokov po smrti autora).

b3 Rozdeľte zariadenia podľa vrstvy komunikačného modelu, na ktorej pracuje

Fyzická vrstva:

Opakovače (repeatery)

Huby (hubs)

Kabely (koaxiálne káble, vlákna, ethernetové káble)

Konvertory medzi rozhraním Ethernet a vláknom (media converters)

Sieťové karty

Linková (dátová) vrstva (Datalinková vrstva):

Prepínače (switches)

Mosty (bridges)

Sieťové karty

Ethernetové karty

Sieťová vrstva:

Routery

IP telefóny

VDSL modemy

Firewally

Transportná vrstva:

TCP/IP zariadenia

Porty

Aplikačná vrstva:

Servery

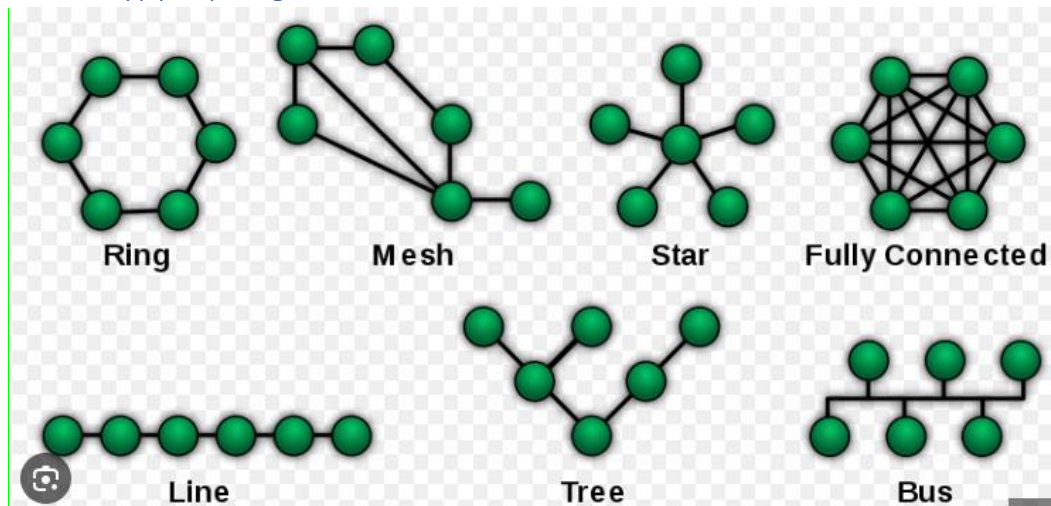
Počítače

Tlačiarne

Telefóny

Smart zariadenia (napr. Smart TV)

c1 Načrtnite typy topológie sietí



c2 Charakterizujte daňovú sústavu, druhy dani a vysvetlite vzťah medzi štátnym rozpočtom a daňami

Daňová sústava je súhrn všetkých zákonov, pravidiel a postupov, ktoré určujú, ako sú získavané, vymáhané a spravované dane vo vnútroštátnom kontexte. Charakterizuje spôsob, akým vláda zhromažďuje príjmy na financovanie svojich činností a poskytuje verejné služby a sociálne výhody.

Príjmové dane: Tieto dane sa zrážajú zo ziskov fyzických osôb (napríklad mzdy), alebo zo ziskov právnických osôb (napríklad zisky podnikov). Príjmové dane môžu byť progresívne alebo regresívne v závislosti od výšky príjmu.

Spotrebné dane: Tieto dane sa platia pri nákupe tovaru alebo služieb a môžu byť zahrnuté v cene tovaru alebo služby (napríklad daň z pridanej hodnoty) alebo môžu byť vyberané samostatne (napríklad daň z alkoholu, tabaku).

Majetkové dane: Tieto dane sa platia na majetok vlastnený jednotlivcami alebo firmami, ako sú napríklad nehnuteľnosti alebo finančné aktíva.

Dedičné dane: Tieto dane sa platia zo zdedeného majetku po úmrtí osoby.

Dane z príjmu z kapitálových výnosov: Tieto dane sa platia z peňazí získaných zo zisku z investícií alebo predaja majetku.

5. Elektronika – Polovodiče - Kruťo

a)

1. Rozdeľte materiály z hľadiska elektrickej vodivosti
2. Vysvetlite vznik polovodiča typu N a polovodiča typu P
3. Popíšte princíp činnosti PN priechodu v priepustnom aj závernom smere
4. Porovnajte vzájomne aspoň 3 typy polovodičových diód a ich použitie
5. Nakreslite časový priebeh harmonického striedavého signálu po jednocestnom a dvojcestnom usmernení

b)

1. Porovnajte princíp činnosti polovodičovej usmerňovacej diódy, tyristora a triaku
2. Popíšte zapojenie NPN tranzistora ako spínača
3. Porovnajte vzájomne aspoň 3 triedy nízkofrekvenčných zosilňovačov

c)

1. Popíšte nastavenie redundancie na L2
2. Popíšte druhy živností a uveďte všetky podmienky nevyhnutné pre ich prevádzkovanie

a1 Rozdeľte materiály z hľadiska elektrickej vodivosti

Vodiče:

Vodiče majú vysokú elektrickú vodivosť.

V tomto type materiálov majú elektróny vo valenčnom pásme väčšiu pohyblivosť, čo umožňuje voľný pohyb nábojov.

Príklady vodičov zahŕňajú kovy ako meď, striebro, zlato a hliník.

Elektróny sa voľne pohybujú v kryštalickej mriežke, môže ľahko vystúpiť z valenčnej sféry. Má množstvo voľných elektrónov.

Izolanty:

Izolanty majú nízku elektrickú vodivosť.

V tomto type materiálov sú elektróny tesne viazané na atómy a nemajú dostatočnú voľnú pohyblivosť.

Príklady izolantov zahŕňajú sklo, keramiku, drevo a plast.

Elektróny sú zaháčknuté, pevne zviazané, preto nevedú prúd. Má strašne silné väzby.

Polovodiče:

Polovodiče majú strednú elektrickú vodivosť medzi vodičmi a izolantmi.

V tomto type materiálov sú elektróny vo valenčnom pásme menej tesne viazané a majú určitú pohyblivosť, ktorá sa môže meniť v závislosti od vonkajších faktorov ako teplota alebo pridané nečistoty.

Príklady polovodičov zahŕňajú kremík, germanium a rôzne polovodičové zlúčeniny ako napríklad kremík a germanium.

Má vlastnosti aj vodičov aj izolantov. Elektróny sú zviazané ako izolant, ale má slabé väzby. Trhajú sa a potom začína tiecť prúd, stačí na to malá energia. Polovodič je materiál, ktorý sa tvári ako izolant, ale pri nepatrnej zmene energie (dodanie energie) sa stáva vodivý.

Prvky: 4. skupina Mendelejevovej tabuľky: kremík, germánium (vo valenčnej sfére sú 4 elektróny.

Prímеси: 5. skupina: polovodič typu N (fosfor, 5 elektrónov vo valenčnej sfére), 3. skupina: polovodič typu P

Prvok sa nastreľí do čistej kremíkovej doštičky

a2 Vysvetlite vznik polovodiča typu N a polovodiča typu P

Polovodič typu N:

Vytvorený pridaním donorových prvkov, ako je fosfor, do čistého polovodičového materiálu.

Fosfor dodáva nadbytok elektrónov, ktoré sú zápornými nositeľmi náboja, čím vytvára typ N polovodič s negatívnou polaritou.

Ak v kryštalickej mriežke štvormocného prvku (germánium, kremík), Kremík nahradíme atómami päťmocného prvku (fosfor...).

Polovodič typu P:

Vytvorený pridaním akceptorových prvkov, ako je bor, do čistého polovodičového materiálu.

Bor vytvára diery vo valenčnej vrstve, ktoré sú pozitívnymi nositeľmi náboja, čím vytvára typ P polovodič s pozitívnou polaritou.

Polovodič typu P získame vtedy, ak atóm napr. kremíka z kryštalickej mriežky nahradíme prvkom z 3. skupiny Mendelejevovej tabuľky.

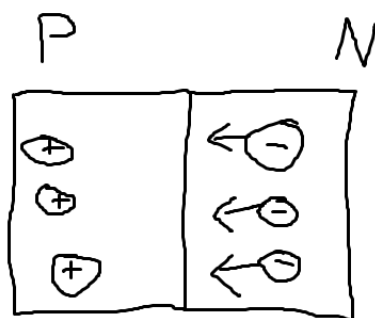
a3 Popíšte princíp činnosti PN priechodu v priepustnom aj závernom smere

PN prechod nám vzniká po spojení polovodiča typu P, N na hranici spojenia. Podmienka: spojenie týchto polovodičov musí byť elektricky vodivé. PN prechod usmerňuje tečenie elektrického prúdu.

Priepustný smer:

Voľné elektróny na N strane sa pohybujú k dierkam na P strane, vytvárajúc malý odpor.

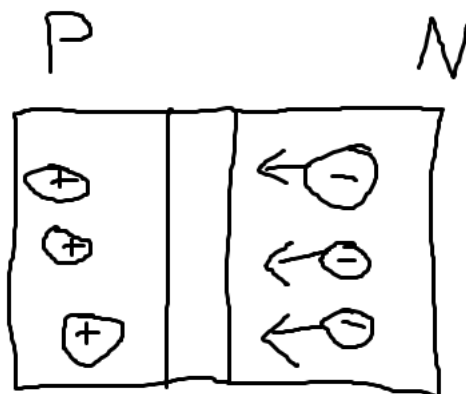
Vytvára sa deplečná vrstva na hranici, ktorá bráni ďalšiemu toku prúdu.



Záverom smer:

Voľné elektróny a diery sú odtiahnuté ďaleko od hranice, rozširuje sa deplečná vrstva.

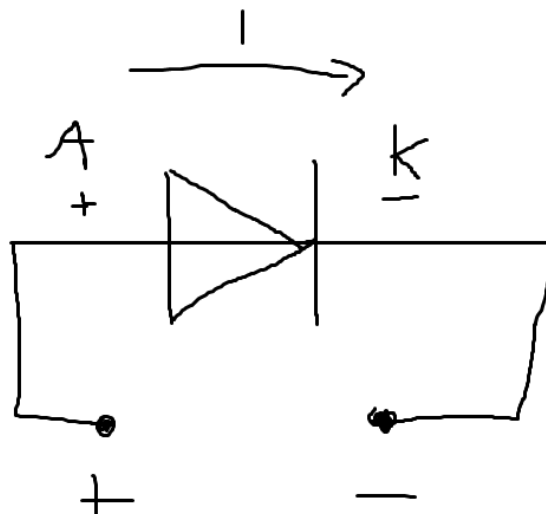
Vysoký odpor sa dosahuje, pretože deplečná vrstva bráni prúdeniu prúdu cez PN prechod.



Asi takto???

a4 Porovnajte vzájomne aspoň 3 typy polovodičových diód a ich použitie

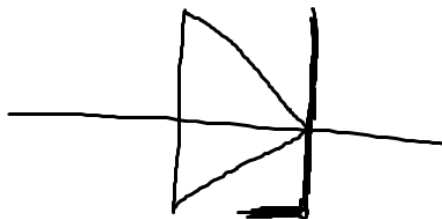
Dióda je polovodičová súčiastka, ktorá má 1 PN prechod.



Zenerova dióda:

Je to dióda, ktorá sa využíva na prácu v závernom smere. Má veľmi strmú VA charakteristiku, čo znamená, že malá zmena napätia vyvolá veľkú zmenu tečenia prúdu.

Používa sa na stabilizáciu napätia v obvode. Keď je aplikované napätie presahujúce určitú hodnotu (napríklad 3,6 V pre BZX55C3V6), dióda začne vodiť, udržiavať tak napätie na konštantnej úrovni.

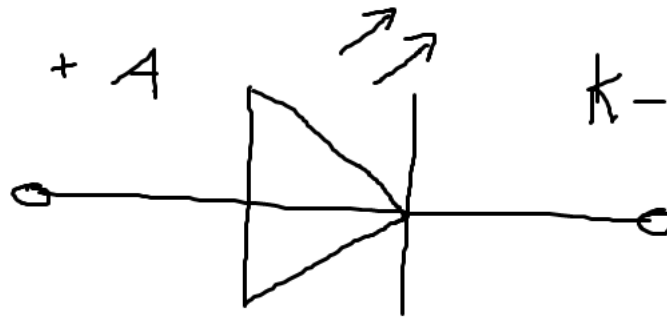


LED dióda (Light Emitting Diode):

LED dióda usmerňuje elektrický prúd. Red, green, blue.

LED diódy pretvárajú elektrickú energiu priamo na svetlo, čím poskytujú vysokú účinnosť a dlhú životnosť.

Sú využívané v osvetlení (domáce osvetlenie, displeje, svetelné reklamy), optických komunikáciách a v rôznych senzorických aplikáciách.



a5 Nakreslite časový priebeh harmonického striedavého signálu po jednocestnom a dvojcestnom usmernení

Text

b1 Porovnajzte princíp činnosti polovodičovej usmerňovacej diódy, tyristora a triaku

Polovodičová usmerňovacia dióda: Pasívny prvok, umožňuje prúd prechádzať len v jednom smere.

Tyristor: Aktívny spínací prvok, umožňuje kontrolu prúdu v jednom smere po zápale.

Triak: Podobný tyristoru, ale môže spínať prúd v oboch smeroch a má dve brány pre riadenie prúdu.

b2 Popíšte zapojenie NPN tranzistora ako spínača

Zapojenie NPN tranzistora ako spínača:

NPN Tranzistor: Skladá sa z emitora (E), bázy (B) a kolektora (C).

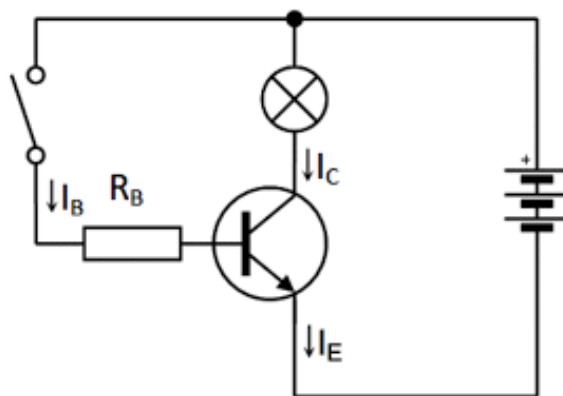
Riadiaci Signál: Napätie alebo prúd aplikovaný na bázu.

Funkcia Spínača:

Vysoký signál na báze spína tranzistor (je zatvorený).

Nízky alebo žiadny signál na báze vypne tranzistor (je otvorený).

Aplikácie: Používa sa v spínačoch, prepínačoch, regulátoroch a ďalších obvodoch na riadenie prúdu.



b3 Porovnajte vzájomne aspoň 3 triedy nízkofrekvenčných zosilňovačov

Tranzistorové zosilňovače sú veľmi flexibilné a môžu byť ľahko prispôsobené pre rôzne aplikácie.

Operačné zosilňovače majú veľmi vysoké zosilnenie a sú veľmi presné, čo ich robí vhodnými pre precízne aplikácie.

Diferenčné zosilňovače sú špecifické pre zosilňovanie rozdielných signálov a často sa používajú v situáciách, kde je dôležitý vysoký pomer signál-šum a potlačenie spoločného režimu

c1 Popíšte nastavenie redundancie na L2

Nastavenie redundancie na úrovni L2 sa často využíva v počítačových sieťach na zvýšenie spoľahlivosti a odolnosti voči výpadkom. Jedným z najčastejších spôsobov, ako dosiahnuť redundanciu na úrovni L2, je použitie technológie nazývanej Spanning Tree Protocol (STP)

Spanning Tree Protocol (STP):

STP je protokol na úrovni L2, ktorý eliminuje slučky v topológiách s viacerými cestami tým, že identifikuje a blokuje zbytočné spojenia.

V rámci STP sa vyberie jeden "koreňový" most, ktorý bude mať najnižšiu prioritu, a potom sa pre každý most na sieti vypočíta najkratšia cesta k tomuto koreňovému mostu.

Ak existujú viaceré cesty medzi dvoma bodmi v sieti, STP blokuje všetky okrem jednej, čím sa zabráni tvorbe slučiek a kolízií v sieti.

c2 Popíšte druhy živností a uveďte všetky podmienky nevyhnutné pre ich prevádzkovanie

Živnostník (fyzická osoba):

Podmienky: Živnostenský list, identifikačný doklad, splnenie požiadaviek pre konkrétnu živnosť.

Charakteristika: Jednotlivec prevádzkujúci podnikateľskú činnosť s osobnou zodpovednosťou.

Verejná obchodná spoločnosť (VOS):

Podmienky: Viacerí spoločníci, stanovenie spoločenskej zmluvy, registrácia v Obchodnom registri.

Charakteristika: Spoločnosť s viacerými spoločníkmi, zdieľajúcimi zodpovednosť za záväzky.

Spoločnosť s ručením obmedzeným (s.r.o.):

Podmienky: Minimálne jedného spoločníka, založenie spoločenskej zmluvy, registrácia v Obchodnom registri.

Charakteristika: Spoločnosť s jedným alebo viacerými spoločníkmi, s obmedzenou zodpovednosťou.

Akciová spoločnosť (a.s.):

Podmienky: Minimálny základný imanžel, vypracovanie stanov, registrácia v Obchodnom registri.

Charakteristika: Spoločnosť s vlastníctvom rozdeleným na verejne obchodovateľné akcie, s oddelenou zodpovednosťou akcionárov.

6. Sieťové – Komunikačné modely - Mišo

a)

1. Definujte vrstvový komunikačný model OSI a model TCP/IP
2. Popíšte spôsob komunikácie medzi jednotlivými vrstvami modelu, použite obrázok 1 obrazovej prílohy č.6
3. Popíšte proces encapsulácie a deencapsulácie
4. Vysvetlite pojem PDU, použite obrázok 2 obrazovej prílohy č. 6
5. Porovnajte hlavičky PDU Frame a Packet

b)

1. Zdôvodnite potrebu použitia verejných a privátnych IP adries v IPv4
2. Uveďte názvy aspoň dvoch organizácií zodpovedných za otvorené štandardy v sieťach
3. Vysvetlite využitie protokolu NAT

c)

1. Uveďte výhody a nevýhody jednotlivých zdrojov, metód získavania nových zamestnancov a popíšte základné náležitosti pracovnej zmluvy
2. Vysvetlite pojem algoritmus z hľadiska programovania a vymenujte jeho základné vlastnosti

a1 Definujte vrstvový komunikačný model OSI a model TCP/IP

Model OSI (Open Systems Interconnection):

Model OSI je teoretický rámec navrhnutý ako základná príručka pre návrh a implementáciu komunikačných systémov, ktoré sú otvorené pre interakciu a interoperabilitu medzi rôznymi výrobcami a systémami. Model OSI je rozdelený do sedmi taktových vrstiev:

1.Fyzická vrstva (Physical Layer):

- Definuje fyzické parametre pre prenos signálov a dáta medzi zariadeniami, ako sú napríklad káble, konektory a frekvencie.

2. Linková vrstva (Data Link Layer):

- Zabezpečuje spoľahlivú komunikáciu medzi zariadeniami na rovnakej fyzickej sieti, identifikuje chyby a opravuje ich, ak je to možné.

3. Sieťová vrstva (Network Layer):

- Riadi smerovanie dát medzi rôznymi sieťami a definuje logické adresy (IP adresy) pre identifikáciu zariadení a smerovanie dát.

4. Transportná vrstva (Transport Layer):

- Zabezpečuje spoľahlivú komunikáciu medzi aplikáciami na rôznych zariadeniach, kontroluje tok dát a rieši chyby v doručovaní.

5. Relačná vrstva (Session Layer):

- Umožňuje a riadi komunikáciu medzi aplikáciami pomocou úvodných, udržiavacích a ukončovacích relácií.

6. Presentation Layer:

- Zabezpečuje interoperabilitu medzi rôznymi formátmi dát a zabezpečuje, že dáta sú vhodné pre prenos a interpretáciu.

7. Application Layer:

- Poskytuje rozhranie pre používateľské aplikácie a služby a umožňuje im prístup k sieťovým službám ako je e-mail, prehliadače, sťahovanie súborov atď.

Model TCP/IP (Transmission Control Protocol/Internet Protocol):

Model TCP/IP je praktický rámec, ktorý popisuje, ako sú dáta prenášané cez internet. Skladá sa z štyroch základných vrstiev:

1. Vrstva sieťového rozhrania (Network access):

- Obdobná linkovej vrstve modelu OSI, zabezpečuje prenos dát medzi zariadeniami v rámci jednej fyzickej siete.

2. Internetová vrstva (Internet Layer):

- Podobná sieťovej vrstve modelu OSI, zabezpečuje smerovanie dát medzi rôznymi sieťami a zariadeniami prostredníctvom IP adres.

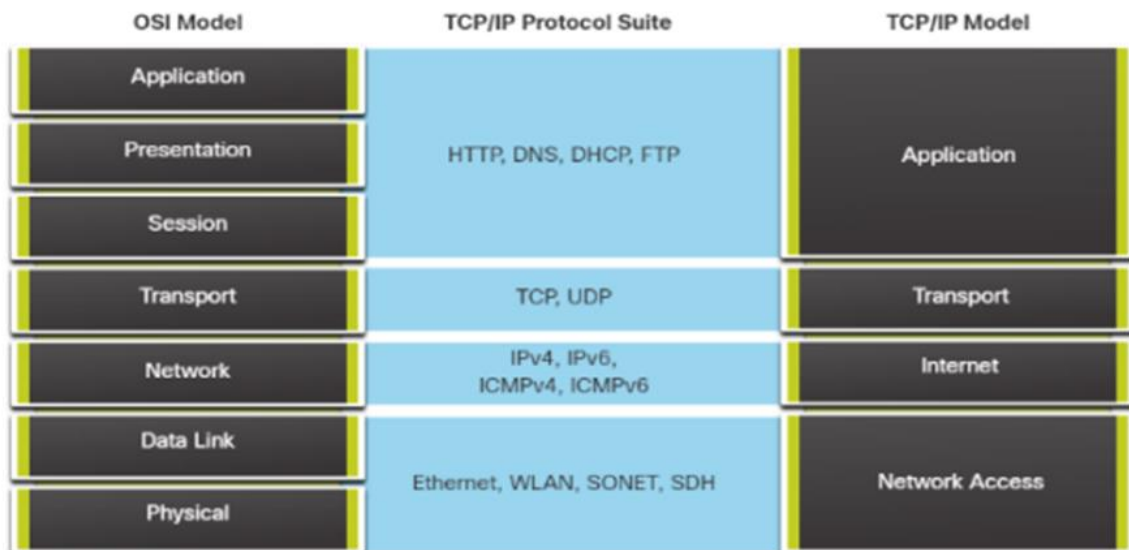
3. Transportná vrstva (Transport Layer):

- Zodpovedá transportnej vrstve modelu OSI, zabezpečuje spoľahlivú komunikáciu medzi aplikáciami na rôznych zariadeniach pomocou protokolov ako TCP a UDP.

4. Aplikačná vrstva (Application Layer):

- Podobne ako aplikačná vrstva modelu OSI, poskytuje rozhranie pre používateľské aplikácie a umožňuje im prístup k rôznym sieťovým službám.

Hlavný rozdiel medzi modelmi spočíva v tom, že model OSI je teoretický rámec, ktorý poskytuje štruktúru a základy pre štandardizáciu komunikácie, zatiaľ čo model TCP/IP je praktický rámec, ktorý je široko používaný vo svete sietí a internetu.



a2 Popíšte spôsob komunikácie medzi jednotlivými vrstvami modelu, použite obrázok 1 obrazovej prílohy č.6

Každá vrstva v modeli OSI komunikuje s okolitými vrstvami prostredníctvom definovaných protokolov a rozhraní. Každá vrstva zodpovedá za určitú úlohu a poskytuje určité služby vrstvám vyššie a nižšie v rámci komunikačného procesu. Komunikácia medzi vrstvami prebieha pomocou jednotiek dát nazývaných PDU (Protocol Data Units), ktoré sa prenášajú od jednej vrstvy k druhej.

1. Aplikačná vrstva (Application Layer):

Komunikuje s aplikáciami a používateľským rozhraním. Poskytuje služby a protokoly, ktoré umožňujú aplikáciám prístup k sieťovým zdrojom. Príklady komunikačných protokolov zahŕňajú HTTP, FTP, SMTP a DNS.

2. Prezentačná vrstva (Presentation Layer):

Stará sa o formátovanie, kompresiu a šifrovanie dát, aby boli vhodné pre prenos cez sieť. Môže transformovať dáta do štandardných formátov, aby boli kompatibilné s rôznymi systémami. Príkladom môže byť prevod textu z jedného kódovania do iného.

3. Relačná vrstva (Session Layer):

Umožňuje, riadi a ukončuje relácie medzi aplikáciami. Zodpovedá za správu spojení a synchronizáciu komunikácie medzi dvoma systémami. Príkladom môže byť zahájenie, udržiavanie a ukončenie spojenia medzi webovým klientom a serverom.

4. Transportná vrstva (Transport Layer):

Zodpovedá za spoľahlivý prenos dát medzi zariadeniami. Poskytuje služby kontroly toku, segmentácie, detekcie chýb a opravy chýb. Protokoly na tejto úrovni zahŕňajú TCP (Transmission Control Protocol) a UDP (User Datagram Protocol).

5. Sieťová vrstva (Network Layer):

Riadi smerovanie dát medzi zariadeniami v rôznych sieťach. Používa logické adresy (IP adresy) na identifikáciu zariadení a rozhoduje o najvhodnejšej ceste prenosu dát. Príkladom je protokol IP (Internet Protocol).

6. Linková vrstva (Data Link Layer):

Poskytuje spoľahlivý prenos dát medzi susednými zariadeniami v rámci rovnakej fyzickej siete. Zabezpečuje detekciu a opravu chýb na fyzickej úrovni a riadi prístup k médiu. Príklady zahŕňajú protokoly Ethernet, WiFi a PPP.

7. Fyzická vrstva (Physical Layer):

Zabezpečuje prenos signálov a dát cez fyzické médiá, ako sú káble, optické vlákna alebo rádiové vlny. Definuje elektrické, optické a mechanické vlastnosti prenosového média. Príklady zahŕňajú protokoly Ethernet, USB a Wi-Fi.

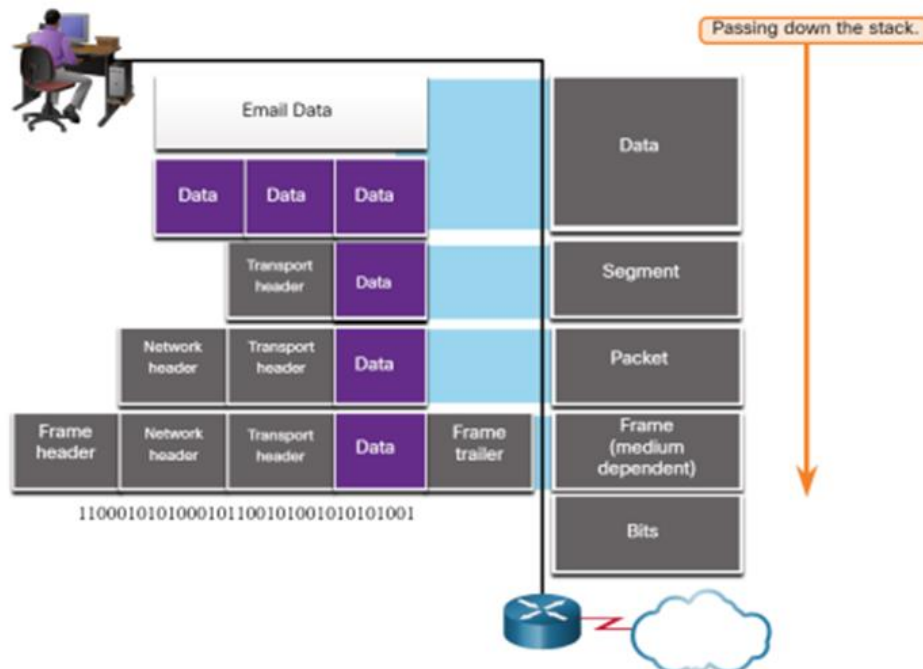
a3 Popíšte proces encapsulácie a deencapsulácie

Encapsulácia:

Encapsulácia je proces pridávania hlavičiek a prípadne aj pätičiek k dátam, aby sa vytvorila správa alebo rámec na odoslanie cez sieť. Pri komunikácii v sieti sú dáta postupne balené do rôznych vrstiev hlavičiek, pričom každá vrstva pridáva svoju vlastnú hlavičku.

Napríklad, pri odosielaní dát pomocou **TCP/IP** modelu:

1. Aplikačná vrstva pridáva dáta a vytvára segment.
2. Transportná vrstva pridáva hlavičku transportného protokolu (napr. TCP alebo UDP) k segmentu a vytvára segment alebo datagram.
3. Sieťová vrstva pridáva hlavičku IP adresy (napr. IPv4 alebo IPv6) k segmentu alebo datagramu a vytvára paket.



4. Linková vrstva pridáva hlavičku a prípadne aj päťu (napr. Ethernetový rámec) k paketu a vytvára rámec.

5. Fyzická vrstva prevádza rámec na fyzické signály a odosiela ich cez médium.

Deencapsulácia:

Deencapsulácia je opačný proces, ktorý sa uskutočňuje na strane prijímateľa. Pri prijímaní dát sú vrstvy hlavičiek postupne odstraňované, pričom každá vrstva kontroluje správnosť svojich dát a prenáša ich na vyššiu vrstvu.

Pokračujúc v príklade odosielania dát pomocou TCP/IP modelu:

1. Fyzická vrstva prijíma signály a prevádza ich na rámec.
2. Linková vrstva prijíma rámec, overuje jeho integritu a odstraňuje hlavičku a prípadne aj päť.
3. Sieťová vrstva prijíma paket, overuje jeho cieľovú IP adresu a odstraňuje hlavičku.
4. Transportná vrstva prijíma segment alebo datagram, overuje protokol a odstraňuje hlavičku.
5. Aplikačná vrstva prijíma dáta a spracováva ich podľa aplikácie.

a4 Vysvetlite pojem PDU, použite obrázok 2 obrazovej prílohy č. 6

PDU (Protocol Data Unit) je základná jednotka údajov, ktorá je prenášaná medzi vrstvami v modeli komunikácie, ako je model OSI alebo TCP/IP. PDU je "obálka" alebo "balík" údajov, ktorý obsahuje samotné dáta, ako aj metadáta (napríklad hlavičky a prípadne aj päť), ktoré sú potrebné pre ich prenos cez sieť. Každá vrstva modelu komunikácie pridáva alebo odstraňuje svoju vlastnú PDU počas procesu encapsulácie a deencapsulácie.

- Na obrázku 2 môžeme vidieť, ako je údajový prúd rozdelený do jednotlivých PDU v rámci rôznych vrstiev modelu OSI. Každá vrstva pridáva svoju vlastnú hlavičku a/alebo päť k dátam, čím vytvára kompletnú PDU, ktorá je prenášaná cez sieť.
- Napríklad, v rámci TCP/IP modelu by PDU na úrovni transportnej vrstvy mohla byť nazývaná segment (pre TCP) alebo datagram (pre UDP). Tieto segmenty alebo datagramy obsahujú samotné dáta, ako aj informácie o zdroji a cieľových portoch (pre TCP) alebo kontrolné informácie (pre UDP).

a5 Porovnajte hlavičky PDU Frame a Packet

Text

b1 Zdôvodnite potrebu použitia verejných a privátnych IP adries v IPv4

1. Verejné IP adresy:

- Verejné IP adresy sú adresy, ktoré sú globálne unikátne a priradené k zariadeniam pripojeným priamo k internetu.
- Tieto adresy sú používané na identifikáciu jednotlivých zariadení vo verejnej časti internetu a umožňujú im komunikovať medzi sebou cez internet.
- Ich použitie je nevyhnutné pre správne fungovanie internetu a poskytuje jednoznačnú identifikáciu zariadení.

2. Privátne IP adresy:

- Privátne IP adresy sú adresy, ktoré sú používané v súkromných sieťach (napríklad domáce siete, firemné intranety).
- Tieto adresy nie sú unikátne a môžu byť opakovane použité vo viacerých súkromných sieťach.

- Použitie privátnych adries umožňuje organizáciám vytvárať vlastné uzavreté siete, ktoré sú chránené pred priamym prístupom z internetu.
- Privátne IP adresy sú rezervované pre použitie v súkromných sieťach a nemusia byť jednoznačné na celom internete.

Zdôvodnenie potreby použitia oboch typov adries spočíva v ich unikátnych vlastnostiach a zabezpečení funkčnosti a bezpečnosti internetu a súkromných sietí. Verejné IP adresy sú nevyhnutné pre identifikáciu zariadení na internete, zatiaľ čo privátne IP adresy poskytujú možnosť vytvárať uzavreté siete s obmedzeným prístupom zvonku. Ich kombinované použitie umožňuje efektívne fungovanie celého internetového ekosystému.

b2 Uved'te názvy aspoň dvoch organizácií zodpovedných za otvorené štandardy v sieťach

Internet Engineering Task Force (IETF):

- IETF je mimovládna organizácia zameraná na štandardizáciu internetových technológií a protokolov.
- Je známa svojimi sériami RFC (Request for Comments), ktoré popisujú štandardy a protokoly používané na internete.
- IETF pracuje na vývoji a údržbe mnohých kľúčových protokolov, ako sú TCP/IP, HTTP, TLS, IPv6 a mnoho ďalších.

Institute of Electrical and Electronics Engineers (IEEE):

- IEEE je technická profesijná organizácia, ktorá sa zaoberá štandardizáciou v rôznych odvetviach, vrátane informačných technológií a sieťových technológií.
- IEEE má široký vplyv na štandardizáciu sietí, vrátane LAN (Local Area Network) a bezdrôtových technológií ako WiFi.
- Medzi známe štandardy patrí napríklad Ethernet (IEEE 802.3) alebo bezdrôtové siete (IEEE 802.11).

b3 Vysvetlite využitie protokolu NAT

Protokol NAT (Network Address Translation) sa využíva na prekladanie adries medzi rôznymi sieťovými doménami. Hlavným cieľom NAT je umožniť viacerým zariadeniam v sieti, ktoré sú skryté za jedným verejným IP adresou, komunikovať s internetom.

- 1. Použitie súkromných IP adries:** Súkromné IP adresy sú rozsahy adries určené pre použitie v súkromných sieťach, ako sú domáce siete alebo podnikové intranety. NAT umožňuje súkromným zariadeniam komunikovať s verejnými zariadeniami na internete, pričom ich súkromné adresy sú skryté za jednou verejnou adresou.
- 2. Zabezpečenie siete:** NAT môže slúžiť ako forma zabezpečenia siete tým, že skrýva interné IP adresy za verejnou adresou. To môže znemožniť priamemu prístupu k interným zariadeniam zo zvonku siete, čím sa znižuje riziko útokov z internetu.
- 3. Ekonomické využitie verejných adries:** Vzhľadom k obmedzenému množstvu verejných IPv4 adries môže byť drahé priradiť každému zariadeniu vo firme alebo domácej sieti svoju jedinečnú verejnú adresu. NAT umožňuje viacerým zariadeniam využívať jednu verejnú adresu, čo môže znížiť náklady na využitie verejných adries.

4. Riadenie toku dát: NAT môže byť použitý na riadenie toku dát v sieti tým, že filtruje alebo modifikuje dáta, ktoré prechádzajú cez neho. To môže byť užitočné pre rôzne účely, ako sú filtrovanie obsahu, zvýšenie bezpečnosti siete alebo optimalizácia siete.

c1 Uvedte výhody a nevýhody jednotlivých zdrojov, metód získavania nových zamestnancov a popíšte základné náležitosti pracovnej zmluvy

Získavanie zamestnancov je činnosť zameraná na získanie vhodných uchádzačov o voľné pracovné miesto.

Zdroje získavania zamestnancov:

- **Externé zdroje** – ide o uchádzačov o zamestnanie z vonkajšieho prostredia. Patria sem absolventi škôl, vzdelávacích kurzov, voľné pracovné sily na trhu práce (nezamestnaní evidovaní na úradoch práce), zamestnanci iných podnikov, ktorí si hľadajú nové zamestnanie a podobne.
- Výhodou externých zdrojov je širšia možnosť výberu zamestnancov, možnosť využitia skúseností zamestnancov z iných podnikov, nové myšlienky a nápady na riešenie podnikových problémov.
- Nevýhodou externých zdrojov sú vyššie náklady na získavanie zamestnancov, dlhší čas potrebný na výber uchádzačov, neznalosť podniku a tým dlhšia doba zapracovania zamestnancov (adaptácie), prenášanie zlovykov z bývalého pracoviska a podobne.
- **Interné zdroje** – ide o vlastných zamestnancov podniku, ktorí môžu postúpiť na voľné pracovné miesto alebo môžu byť priradení na základe plánu rozmiestnenia zamestnancov.
- K výhodám interných zdrojov patria nižšie náklady na získavanie zamestnancov, rýchlejšie obsadenie pracovného miesta, možnosť pracovného postupu zamestnancov, vyššia motivácia zamestnancov a podobne.
- Nevýhodou sú vyššie náklady na vzdelávanie zamestnancov, obmedzený výber uchádzačov, prehliadanie chýb vlastných zamestnancov, chýbajú nové pohľady na riešenie podnikových problémov a podobne.

Metódy získavania zamestnancov:

Podnik využíva na získavanie zamestnancov viaceré metódy. K najčastejším patria:

- inzercia v médiách – denná tlač, inzertné noviny, časopisy,
- internet – zverejnenie ponuky cez vlastné internetové stránky a internetové portály,
- spolupráca so sprostredkovateľskými a personálnymi agentúrami,
- vlastná databáza (uchádzačov, bývalých zamestnancov),
- plagáty, letáky, vývesky,
- sprostredkovanie cez úrady práce,
- spolupráca so školami, vzdelávacími agentúrami,
- pracovné veľtrhy a burzy práce,
- odporúčania (tzv. referencie) od súčasných zamestnancov, prípadne známych,
- personálny lízing a iné.

Základné náležitosti pracovnej zmluvy:

- **Identifikácia zamestnávateľa a zamestnanca:** Zmluva musí jasne uvádzať mená a kontaktné údaje zamestnávateľa a zamestnanca.
- **Pracovné podmienky:** Zmluva by mala stanoviť pracovnú pozíciu, pracovný čas, miesto práce a ostatné dôležité pracovné podmienky.
- **Platy a odmeny:** Zmluva by mala určiť výšku mzdy alebo platového pásma, spôsob platby, prípadné bonusy a ďalšie odmeny.
- **Pracovné povinnosti:** Zmluva by mala stanoviť jasné povinnosti a očakávania zo strany zamestnávateľa a zamestnanca.
- **Dĺžka zmluvy:** Zmluva by mala určiť dátum začiatku a konca zamestnania, ak ide o dočasnú alebo časovo obmedzenú zmluvu.
- **Práva a povinnosti strán:** Zmluva by mala definovať práva a povinnosti zamestnanca a zamestnávateľa vrátane práva na dovolenku, zdravotné poistenie, ochranu osobných údajov atď.
- **Podmienky ukončenia pracovného pomeru:** Zmluva by mala obsahovať podmienky ukončenia pracovného pomeru zo strany zamestnanca aj zamestnávateľa, vrátane možných odstúpeného, výpovedných lehôt a iných dôležitých podmienok.

c2 Vysvetlite pojem algoritmus z hľadiska programovania a vymenujte jeho základné vlastnosti

V kontexte programovania je **algoritmus** presný postup alebo séria krokov, ktoré sú napísané v podobe počítačového kódu. Tieto kroky určujú, ako má počítač vykonávať určitú úlohu alebo riešiť určitý problém. Algoritmy v programovaní môžu byť napísané v rôznych programovacích jazykoch a používajú sa na implementáciu funkcionality softvéru. Je to vlastne daný postup, ktorý na základe vstupných údajov dosiahne nejaký výstup.

- **Hromadnosť** – Opakuje rovnakú úlohu x krát, dá sa použiť na viac napríklad príkladov
- **Efektívnosť** – má zabezpečiť riešenie v čo najkratšom čase, bude tam čo najmenej inštrukcií.
- **Elementárnosť (jednoduchosť)** – chceme čo najjednoduchšie kroky
- **Determinovanosť** – po každom kroku vieme povedať, aký ďalší krok nasleduje
- **Konečnosť** – skončí za rozumný čas
- **Rezultatívnosť (presnosť)** – výsledok algoritmu by mal byť čo najpresnejší. Síce sa mi ten program spustil, ale nevyhodil požadovaný výsledok. Napr. $5 * (-6) = 30$ a má to byť -30 .

7. Sieťové – Bezpečnosť na sieti - Mišo

a)

1. Definujte pojem bezpečnosť počítačovej siete
2. Vysvetlite možné hrozby počítačovej siete
3. Popíšte autentifikáciu a autorizáciu
4. Vysvetlite možnosti VLAN
5. Popíšte útoky DHCP

b)

1. Zdôvodnite potrebu zabezpečenia sieťových zariadení
2. Načrtnite spôsoby zabezpečenia siete

3. Aplikujte spôsoby zabezpečenia dát

c)

1. Uvedte základné marketingové nástroje, stratégie využívané v podniku a znázornite životný cyklus výroby
 2. Definujte pojem premenná v programovaní, predvedte a vysvetlite deklaráciu premennej
-

a1 Definujte pojem bezpečnosť počítačovej siete

Bezpečnosť počítačovej siete sa odvoláva na súbor opatrení, postupov a technológií, ktoré sú navrhnuté a implementované na ochranu počítačových sietí pred neoprávneným prístupom, útokmi, poškodením alebo zneužitím. Cieľom bezpečnostných opatrení je zabezpečiť, aby informácie a zdroje v počítačovej sieti zostali dôverné, celistvé a dostupné pre oprávnených používateľov.

1. **Dôvernosc:** Zabezpečuje, že citlivé informácie a údaje v sieti sú dostupné len oprávneným používateľom. To sa dosahuje prostredníctvom kryptografických techník, autentifikácie a prístupových kontrol.
2. **Celočíselnosc:** Zabezpečuje, že údaje v sieti zostávajú nedotknuté a neupravené počas prenosu. Mechanizmy na zabezpečenie celočíselnosti zahŕňajú použitie kryptografických hašovacích funkcií a digitálnych podpisov.
3. **Dostupnosť:** Zabezpečuje, že služby a zdroje v počítačovej sieti sú dostupné pre oprávnených používateľov v požadovanej miere. Ochrana pred útokmi typu DoS (Denial of Service) a zálohovanie údajov sú dôležité súčasti zabezpečenia dostupnosti.
4. **Integrita:** Zabezpečuje, že systémy a údaje v sieti zostávajú integrované a dôveryhodné. To zahŕňa použitie digitálnych podpisov a kontroly prístupu na zabránenie neoprávnených zmenám alebo úpravám.
5. **Autentifikácia:** Overuje identitu používateľov a zariadení, ktoré sa pripájajú k sieti. Autentifikácia môže byť dosiahnutá pomocou hesiel, biometrických údajov, certifikátov alebo dvojfaktorovej autentifikácie.
6. **Auditoria:** Zabezpečuje sledovanie a zaznamenávanie udalostí v sieti na účely monitorovania, analýzy a vyšetrovania incidentov. Tieto auditovateľné záznamy pomáhajú identifikovať bezpečnostné hrozby a incidenty.
7. **Firewall:** Firewall je bezpečnostný prvok, ktorý monitoruje a kontroluje tok údajov medzi vnútornými a vonkajšími sieťami. Pomáha zabrániť neoprávnenému prístupu a ochrániť interné zdroje pred externými hrozbami.

a2 Vysvetlite možné hrozby počítačovej siete

Hrozby počítačovej siete predstavujú potenciálne nebezpečenstvá, ktoré môžu ohroziť integritu, dôvernosc a dostupnosť údajov a zdrojov v sieti. Tieto hrozby môžu pochádzať z rôznych zdrojov a môžu mať rôzne formy.

- **Malware:** Malware je široký termín, ktorý zahŕňa rôzne druhy škodlivého softvéru, ako sú vírusy, červy, trojské kone, ransomware a spyware. Tieto programy môžu infikovať počítače a siete, čo vedie k poškodeniu údajov, špionáži, vydieraniu alebo vyčerpaniu zdrojov.
- **Kybernetické útoky:** Kybernetické útoky môžu zahŕňať rôzne formy, ako sú útoky typu DoS (Denial of Service), DDoS (Distributed Denial of Service), útoky na vyčerpanie zdrojov, útoky na

odpojenie, útoky na odpojenie služby alebo únik údajov. Tieto útoky môžu mať za následok nedostupnosť služieb, stratu údajov alebo narušenie prevádzky siete.

- **Phishing:** Phishing je spôsob sociálneho inžinierstva, ktorý sa používa na získanie citlivých informácií, ako sú heslá, bankové údaje alebo osobné údaje, od používateľov. Útočníci často vytvárajú falošné e-maily, webové stránky alebo správy, ktoré sa zdajú byť legitímne, aby používatelia odovzdali svoje údaje.
- **Insider Threats:** Hrozby zo strany zamestnancov alebo interných používateľov môžu byť rovnako nebezpečné ako vonkajšie hrozby. Zamestnanci môžu byť zneužívaní na účely špionáže, krádeže údajov alebo sabotáže s cieľom poškodiť firmu.
- **Zraniteľnosti softvéru:** Zraniteľnosti v softvéri alebo operačných systémoch môžu byť zneužitie na získanie neoprávnenej prístupu k počítačovým systémom alebo na spustenie škodlivého softvéru. Útočníci vyhľadávajú tieto zraniteľnosti a využívajú ich na vykonanie útokov.
- **Zneužívanie autentifikačných a autorizačných mechanizmov:** Útočníci môžu pokúšať o zneužitie autentifikačných a autorizačných mechanizmov na získanie neoprávnenej prístupu k systémom alebo údajom. Tento druh útoku môže zahŕňať napríklad útoky na odhad hesiel, únos účtov alebo zneužitie zraniteľností v autentifikačných mechanizmoch.

a3 Popíšte autentifikáciu a autorizáciu

Autentifikácia a autorizácia sú dva základné prvky zabezpečenia, ktoré sa používajú na ochranu prístupu k systémom, údajom a zdrojom v počítačových sieťach. Aj keď tieto pojmy často idú ruka v ruku a často sa zamieňajú, majú rozdielne významy a účely.

Autentifikácia: zistí kto si

- Autentifikácia sa zaoberá overením identity jednotlivca, zariadenia alebo entit, ktoré sa snažia získať prístup k systému alebo zdrojom.
- Cieľom autentifikácie je potvrdiť, že osoba alebo entita, ktorá sa pokúša získať prístup, je skutočne kým tvrdí byť.
- Tento proces sa vykonáva pomocou rôznych faktorov autentifikácie, ako sú heslá, biometrické údaje (napríklad odtlačky prstov alebo rozpoznanie tváre), certifikáty, tokeny alebo dvojfaktorová autentifikácia.

Autorizácia: udelenie prístupu

- Autorizácia sa týka kontroly prístupu k zdrojom alebo službám po autentifikácii používateľa alebo entity.
- Po overení identity autorizácia určuje, čo je používateľ alebo entita oprávnená robiť v systéme alebo s prístupovými zdrojmi.
- Na základe tejto autentifikácie môžu byť používateľom pridelené rôzne úrovne oprávnení alebo prístupové práva, ktoré definujú, ktoré akcie môžu vykonávať.

V súhrne, **autentifikácia** overuje totožnosť jednotlivca alebo entity, ktorá sa snaží získať prístup, zatiaľ čo **autorizácia** určuje, čo má táto totožnosť právo robiť po získaní prístupu. Spoločne tieto dva procesy pomáhajú zabezpečiť, že iba oprávnené osoby alebo entity majú prístup k systémom a zdrojom, čo zvyšuje bezpečnosť a znižuje riziko neoprávnenej prístupu a zneužitia.

a4 Vysvetlite možnosti VLAN

VLAN (Virtual Local Area Network) je technológia, ktorá umožňuje segmentáciu počítačovej siete na logické skupiny alebo podsiete, aj keď sú fyzicky pripojené do jednej fyzickej siete. Táto segmentácia umožňuje efektívnejšie riadenie, zabezpečenie a správu siete.

- **Logická segmentácia siete:** VLAN umožňuje administrátorom rozdeliť fyzickú sieť na logické skupiny na základe rôznych kritérií, ako sú oddelené oddelenia, funkčné skupiny alebo bezpečnostné zóny. Každá VLAN predstavuje samostatnú logickú sieť s vlastnými prístupovými pravidlami a politikami.
- **Zabezpečenie:** VLAN môže byť použitá na izoláciu určitých skupín zariadení od ostatných v sieti. Napríklad môžete mať jednu VLAN pre oddelenie IT, ďalšiu pre oddelenie účtovníctva a ďalšiu pre návštevníkov. Týmto spôsobom môžete obmedziť prístup k citlivým údajom a aplikáciám.
- **Optimalizácia šírky pásma:** VLAN umožňuje správu šírky pásma prostredníctvom izolácie a segmentácie sieťového prevádzky. To znamená, že prevádzka v jednej VLAN nemusí súťažiť o šírku pásma s prevádzkou v inej VLAN, čo môže viesť k lepšej výkonnosti siete.
- **Optimalizácia správy:** VLAN umožňuje lepšiu organizáciu a správu siete, pretože umožňuje administrátorom priradovať zariadenia do logických skupín na základe ich funkcií, oddelení alebo potrieb. To uľahčuje implementáciu politik bezpečnosti, riadenie prístupu a údržbu siete.
- **Rozšírenie siete:** VLAN umožňuje jednoduchšie a efektívnejšie rozširovanie siete bez potreby fyzickej rekonfigurácie. Nové zariadenia alebo oddelenia môžu byť pridelené do existujúcej VLAN alebo vytvorené nové VLAN podľa potreby, čo zjednodušuje riadenie a správu siete.

a5 Popíšte útoky DHCP

DHCP (Dynamic Host Configuration Protocol) je protokol používaný na dynamické pridelenie IP adries, konfiguráciu siete a ďalších sieťových parametrov zariadeniam pripojeným do siete. Aj keď DHCP je užitočný nástroj na spravovanie sietí, existujú rôzne typy útokov, ktoré môžu zneužiť jeho funkcionality.

- **DHCP Spoofing** (Podvrhnutie DHCP): Pri útoku DHCP spoofing útočník odosiela falošné DHCP odpovede na sieťový dotaz DHCP Request. Útočník môže poskytnúť svoju vlastnú konfiguráciu siete, vrátane falošnej IP adresy brány alebo DNS servera, čo môže viesť k presmerovaniu sieťovej prevádzky na útočníkov systém.
- **DHCP Starvation** (Vyhladzovanie DHCP): Pri útoku DHCP starvation útočník neustále odosiela veľké množstvo DHCP požiadaviek na DHCP server. Cieľom je vyčerpať adresný priestor DHCP poolu, čím sa bráni ostatným legitímnym zariadeniam v získaní IP adries.
- **DHCP Rogue Server** (Falošný DHCP server): Pri tomto útoku útočník vytvára falošný DHCP server v sieti, ktorý poskytuje neoprávnenú konfiguráciu siete zariadeniam. Tieto zariadenia môžu byť následne vystavené rôznym bezpečnostným rizikám alebo môžu byť súčasťou botnetu.
- **DHCP Lease Attack** (Útok na prenájom): Pri útoku na prenájom DHCP útočník získava a udržiava veľké množstvo DHCP prenájmov, čím sa vyčerpáva dostupný adresný priestor DHCP servera. To môže spôsobiť, že legitímne zariadenia nedostanú prístup k IP adresám.
- **DHCP Client Attack** (Útok na klienta DHCP): Pri útoku na klienta DHCP útočník úmyselne manipuluje s klientmi DHCP, napríklad ich prenájom zrušením alebo prekonfigurovaním. To môže mať za následok straty pripojenia alebo nežiaduce správanie zariadení v sieti.

b1 Zdôvodnite potrebu zabezpečenia sieťových zariadení

Zabezpečenie sieťových zariadení je kľúčové pre ochranu integrity, dostupnosti a dôvernosti dát a služieb, ktoré prechádzajú cez počítačové siete. Existuje niekoľko dôvodov, prečo je dôležité zabezpečiť tieto zariadenia:

- **Ochrana pred neoprávneným prístupom:** Sieťové zariadenia, ako sú routery, prepínače, brány a firewally, majú prístup k citlivým informáciám a riadia tok dát v sieti. Ich kompromitácia by mohla umožniť útočníkovi získať neoprávnený prístup k údajom alebo zmeniť sieťovú prevádzku, čo by mohlo viesť k úniku dát alebo nedostupnosti služieb.

- **Prevencia kybernetických útokov:** Zabezpečené sieťové zariadenia môžu pomôcť detegovať a zamedziť rôznym formám kybernetických útokov, ako sú útoky typu DoS (Denial of Service), útoky na prečerpanie zdrojov, útoky na vyčerpanie údajov a mnohé ďalšie. Implementácia bezpečnostných politík a mechanizmov môže pomôcť minimalizovať riziká spojené s týmito útokmi.
- **Ochrana integrovanosti a dôveryhodnosti siete:** Sieťové zariadenia zohrávajú kľúčovú úlohu pri zabezpečovaní integrovanosti a dôveryhodnosti siete. Ich kompromitácia by mohla mať vážne dôsledky pre dôveru používateľov a obchodných partnerov voči sieti a spoločnosti ako celku.
- **Ochrana pred zneužitím:** Zabezpečené sieťové zariadenia môžu chrániť pred zneužitím alebo zneužitím zo strany vnútorných alebo externých hrozieb. To zahŕňa ochranu pred neoprávneným prístupom, zneužitím prístupových práv, zneužitím slabých miest v sieti a inými formami zneužitia.
- **Dodržiavanie predpisov a noriem:** Mnohé odvetvia majú predpisy a normy, ktoré vyžadujú ochranu dát a sietí. Zabezpečené sieťové zariadenia sú nevyhnutné na splnenie týchto požiadaviek a na ochranu pred potenciálnymi pokutami a reputačnými škodami spojenými s nedodržiavaním predpisov.

b2 Načrtnite spôsoby zabezpečenia siete

Firewall:

- Implementácia firewallu je jedným z najzákladnejších krokov pri zabezpečovaní siete. Firewally môžu byť hardvérové alebo softvérové a poskytujú kontrolu toku dát medzi vnútornou a vonkajšou sieťou. Pomáhajú blokovať neoprávnený prístup a monitorovať sieťovú prevádzku.

VPN (Virtual Private Network):

- VPN umožňuje bezpečné pripojenie k sieti cez nezabezpečené pripojenia, ako je internet. Používa šifrovanie na zabezpečenie dát prenášaných medzi klientom a sieťou. To je dôležité pre vzdialených pracovníkov alebo pre prístup k sieti z verejných miest.

Antivírusový a antimalware softvér:

- Antivírusový a antimalware softvér pomáha chrániť zariadenia v sieti pred škodlivým softvérom, ako sú vírusy, červy, trojské kone a ďalšie hrozby. Pravidelné aktualizácie a skenovanie sú nevyhnutné na udržanie účinnosti.

Aktualizácie a patche:

- Pravidelné aktualizácie a patche operačných systémov a aplikácií sú kľúčové pre zabezpečenie siete. Tieto aktualizácie opravujú zistené zraniteľnosti a minimalizujú riziko útokov.

Zabezpečenie bezdrôtových sietí:

- Pri použití bezdrôtových sietí je dôležité používať silné šifrovanie (napríklad WPA2) a zabezpečiť sieť heslom. Je tiež dobré používať metódy identifikácie, ako je MAC filtrácia, na obmedzenie prístupu k sieti.

Bezpečnostné politiky a školenia:

- Implementácia bezpečnostných politík a školení zamestnancov je kľúčová pre zabezpečenie siete. Zahrňte pravidlá pre silné heslá, správne používanie zariadení a softvéru, ako aj oboznamovanie s bezpečnostnými hrozbami a postupmi v prípade incidentov.

Audity a monitorovanie:

- Pravidelné audity a monitorovanie siete môžu pomôcť odhaľovať bezpečnostné problémy a neobvyklé aktivity. Použite nástroje na sledovanie siete a udalostí, ktoré vám umožnia identifikovať a riešiť bezpečnostné incidenty včas.

Zálohovanie a obnova dát:

- Zálohovanie dát je dôležité na ochranu pred stratou údajov v prípade havárie, útoku alebo ďalších nepríjemných udalostí. Uistite sa, že máte riadne zálohované dáta a plány na obnovu v prípade potreby.

b3 Aplikujte spôsoby zabezpečenia dát**Šifrovanie dát:**

- Použitie šifrovania dát je jedným z najdôležitejších spôsobov zabezpečenia dát. Šifrovanie umožňuje premeniť citlivé údaje na nečitateľný kód, ktorý je chránený heslom alebo kľúčom. Šifrovanie by sa malo používať pre uložené dáta, prenos dát cez sieť, ako aj na mobilných zariadeniach a úložiskách v cloude.

Zálohovanie dát:

- Pravidelné zálohovanie dát je nevyhnutné pre ochranu pred stratou dôležitých informácií v prípade havárie, útoku alebo ďalších nepríjemných udalostí. Zálohovanie by malo byť automatizované, pravidelné a zabezpečené, s možnosťou obnovy údajov v prípade potreby.

Správa prístupu:

Kontrola prístupu k dátam je dôležitá pre zabezpečenie, že len oprávnení používatelia majú prístup k citlivým informáciám. Použite technológie ako sú prístupové práva, role a skupiny na riadenie prístupu k dátam na základe potreby a zásady najmenšieho privilegia.

Audity a monitorovanie:

- Pravidelné audity a monitorovanie aktivít používateľov môžu pomôcť odhaľovať neoprávnené aktivity alebo prístupy k dátam. Použite nástroje na sledovanie udalostí, auditovanie prístupu k dátam a detekciu anomálií na identifikáciu potenciálnych hrozieb a zraniteľností.

Zabezpečenie sietí a zariadení:

- Implementujte bezpečnostné opatrenia pre siete a zariadenia, ktoré ukladajú a prenášajú dáta. Toto zahŕňa použitie firewallu, antivírusového softvéru, VPN, aktualizácií a ďalších bezpečnostných opatrení na ochranu pred vonkajšími aj vnútornými hrozbami.

Politiky a školenia:

- Vytvorenie a presadzovanie bezpečnostných politík a pravidiel pre správu dát je dôležité pre zabezpečenie a ochranu dát. Poskytnite školenia zamestnancom o bezpečnostných rizikách, správnom používaní systémov a postupoch pre ochranu dát.

Zabezpečené vývojové postupy:

- Pri vývoji a implementácii softvéru a aplikácií je dôležité zahrnúť bezpečnostné opatrenia a praktiky. To zahŕňa zabezpečené programovanie, testovanie bezpečnostných zraniteľností a zabezpečené konfigurácie systémov.

c1 Uvedte základné marketingové nástroje, stratégie využívané v podniku a znázornite životný cyklus výrobku

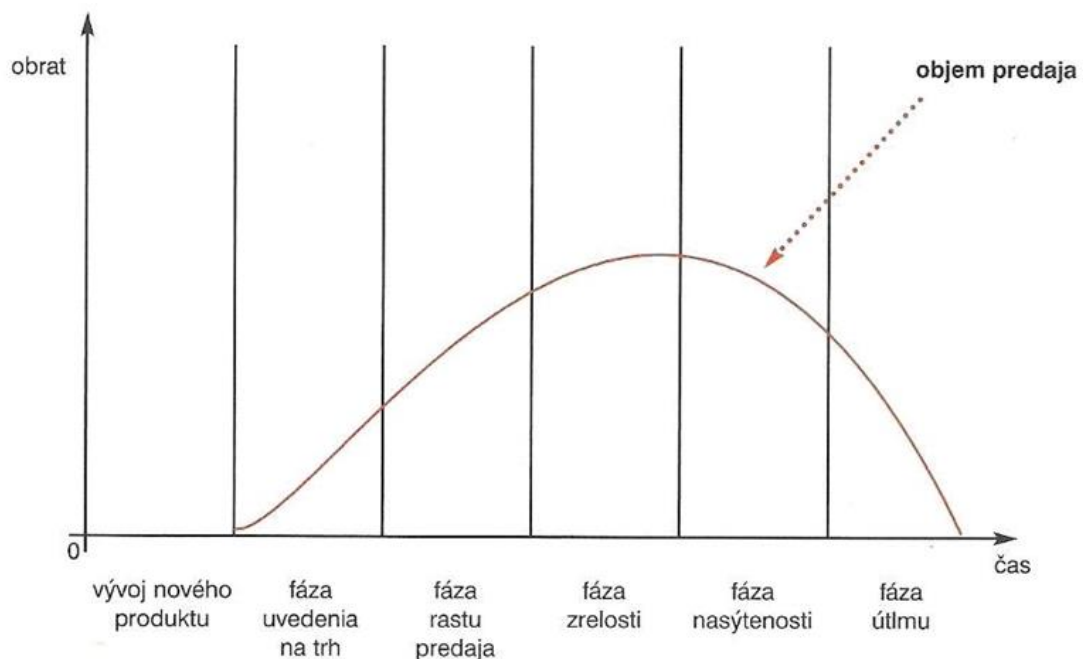
Základné marketingové nástroje zahŕňajú:

- **Produkt:** Definícia produktu alebo služby, ktorú spoločnosť ponúka zákazníkom.
- **Cena:** Stanovenie ceny produktu alebo služby na základe nákladov, konkurencie a hodnoty pre zákazníka.
- **Distribúcia (miesto):** Rozvoj distribučných kanálov a miest, kde je produkt dostupný pre zákazníkov.
- **Propagácia:** Komunikácia s cieľovou skupinou prostredníctvom rôznych marketingových kanálov, ako sú reklamy, sociálne médiá a podpora predaja.

Stratégie využívané v podniku môžu zahŕňať:

- **Segmentácia trhu:** Rozdelenie trhu na skupiny zákazníkov s podobnými potrebami a charakteristikami.
- **Cieľovanie:** Výber jednej alebo viacerých segmentov trhu ako cieľového trhu, na ktorý sa bude zameriavať marketingová aktivita.
- **Pozicionovanie:** Vytvorenie jasného a atraktívneho obrazu produktu alebo značky v mysli zákazníka v porovnaní s konkurenciou.
- **Diferenciácia:** Vytvorenie jedinečnej hodnoty produktu alebo značky, ktorá sa odlišuje od konkurencie.

Životný cyklus produktu:



c2 Definujte pojem premenná v programovaní, predved'te a vysvetlite deklaráciu premennej

Premenná v programovaní je označený miestny úložný priestor, ktorý slúži na uchovávanie hodnôt alebo dát. Táto hodnota alebo dáta môžu byť menené alebo manipulované počas vykonávania programu. Int má 4 bajty (32 bitov)

int cislo = 10; // Deklarácia premennej typu (integer) s názvom 'cislo' a hodnotou 10

8. Sieťové – IP protokoly - Filip

a)

1. Vysvetlite význam a opodstatnenie IP adres
2. Definujte triedy IP adres v IPv4
3. Popíšte význam masky v IPv4
4. Zdôvodnite potrebu broadcastovej adresy
5. Analyzujte potrebu vzniku IPv6

b)

1. Určte možnosti podsieťovania, ak je dĺžka prefixu /24
2. Popíšte vlastnosti IP protokolu
3. Porovnajte hlavičky paketu IPv4 a IPv6, použite obrazovú prílohu č. 8

c)

1. Popíšte rôzne formy platobného styku (elektronické bankovníctvo)
2. Vysvetlite spôsob vykonania príkazu priradenia, uveďte príklad

a1 Vysvetlite význam a opodstatnenie IP adres

IP adresa (Internet Protokol address) – je unikátne/jedinečné číselné označenie (identifikátor) každého zariadenia, ktoré je pripojené k sieti. Pomocou IP adresy identifikujeme zariadenia a správne adresovanie dát (kto/odkiaľ a komu/kam posielal dáta).

2 spôsoby priradovania IP adresy:

Dynamicky – ISP (Internet Services Provider) alebo služba DHCP server/router

Staticky – Manuálne zadanie/priradenie IP adresy zariadeniu

a2 Definujte triedy IP adres v IPv4

IPv4 adresy majú špeciálny 32 bitový číselný formát, ktorý je rozdelený na štyri 8 bitové časti (oktety) oddelené bodkami. Každá z týchto častí je tvorená dekadickým číslom z intervalu 0...255.

Poznáme 5 tried IP adres, každá má iný rozsah možných adres, pri adresovaní používame iba prvé 3

Trieda A

Prvých 8 oktet-ov patrí sieti, zvyšných 24 oktet-ov patrí hosťom, existuje 126 sietí typu A a v každej môže byť až 16 777 214 hostiteľov

IPv4 sieťový rozsah (sú rezervované) od 1.0.0.0/8 do 127.255.255.255/8

IPv4 maska – 255.0.0.0

Prvý hosť/Posledný hosť – [1.0.0.1/8] / [127.255.255.255/8]

Trieda B

Prvých 16 oktet-ov patrí sieti, zvyšných 16 oktet-ov patrí hosťom, existuje 16 384 sietí typu B a v každej môže byť až 65 532 hostiteľov

IPv4 sieťový rozsah od 128.0.0.0/16 do 191.255.255.255/16

IPv4 maska – 255.255.0.0

Prvý hosť/Posledný hosť – [128.0.0.1/16] / [191.255.255.255/16]

Trieda C

Prvých 24 oktet-ov patrí sieti, zvyšných 8 oktet-ov patrí hosťom, existuje 2 097 152 sietí typu C a v každej môže byť až 254 hostiteľov

IPv4 sieťový rozsah od 192.0.0.0/24 do 223.255.255.255/24

IPv4 maska – 255.255.255.0

Prvý hosť/Posledný hosť – [192.0.0.1/24] / [223.255.255.255/24]

Trieda D

Je vyhradená pre skupinovú adresáciu (multicast --> jeden viacerým súčasne)

IPv4 sieťový rozsah od 224.0.0.0 do 239.255.255.255

Trieda E

Je vyhradená pre experimentálne účely

IPv4 sieťový rozsah od 240.0.0.0 do 255.255.255.255

a3 Popíšte význam masky v IPv4

Sieťová maska (Subnet mask) sa využíva na rozdelenie IPv4 adresy na sieťovú a hosťovskú časť.

192.168.1.10/24 (255.255.255.0) --> [192.168.1] reprezentuje **sieťovú časť** a [.10] reprezentuje **hosťovskú časť**

Slúži aj k lepšej prehľadnosti v decimálnom bodkovom zápise (vzor: 192.168.1.10 255.255.255.0)

a4 Zdôvodnite potrebu broadcastovej adresy

Broadcast adresa – slúži na rozposielanie dát všetkým zariadeniam na sieti (jeden všetkým). Využívajú ju napr.:

- Switche keď posielajú packet a nemajú cieľovú MAC-adresu v MAC-adresovej tabuľke,
- Protokol ARP keď zariadenie nemá IPv4 adresu,
- Služba dynamického DHCPv4/DHCPv6, keď si zariadenie pýta IPv4/IPv6 adresu.

a5 Analyzujte potrebu vzniku IPv6

V dobe vytvorenia systému tvorby IP čísla sa nepredpokladalo, že veľké siete budú mať taký rozsah ako dnes. IPv4 (IP verzia 4) adresy sa zapisujú vo formáte **32-bitov**, decimálne a maximálna kapacita sú 4,3 miliardy adries, **ktorých je nedostatok**. Tento problém sa rieši pomocou dynamického prideľovania IP adries, **NAT** (Network Address Translation). Tvorba vytvorenia IP čísla bola nedávno inovovaná na **IPv6** (IP verzia 6). IPv6 adresy sa zapisujú vo formáte **128-bitov**, hexadecimálne a maximálna kapacita sú 340 undecillion adries.

b1 Určte možnosti podsieťovania, ak je dĺžka prefixu /24

Default mask: využitie defaultného /24 subnetu na ktorý je možno pripojiť 256 hostov.

Rozdelenie do menších subnetov: /24 subnet vieme rozdeliť na menšie subnety tým že využijeme/si požičiame niekoľko bitov z hostovskej časti. Napr. 255.255.255.128 = /25 – dve subnety, na každej 128 adries.

Supernetting: je proces kombinovania susedných sietí do väčšej siete. To sa môže vykonať v prípade, že máme viacero sietí s rovnakým prefixom napr. /24, môžeme ich spojiť do jednej väčšej siete s prefixom menším ako /24

Subnet Mask	32-bit Address	Prefix Length
255.0.0.0	11111111.00000000.00000000.00000000	/8
255.255.0.0	11111111.11111111.00000000.00000000	/16
255.255.255.0	11111111.11111111.11111111.00000000	/24
255.255.255.128	11111111.11111111.11111111.10000000	/25
255.255.255.192	11111111.11111111.11111111.11000000	/26
255.255.255.224	11111111.11111111.11111111.11100000	/27
255.255.255.240	11111111.11111111.11111111.11110000	/28
255.255.255.248	11111111.11111111.11111111.11111000	/29
255.255.255.252	11111111.11111111.11111111.11111100	/30

b2 Popíšte vlastnosti IP protokolu

Connectionless: predtým ako odošle dátové pakety, nenadväzuje spojenie s cieľovou destináciou.

Best-Effort doručovanie: IP pracuje na základe „best-effort“, čo znamená, že sa snaží doručiť pakety do cieľovej destinácie čo najefektívnejšie, ale nie je zaručené že sa to podarí. To znamená, že jeho spoľahlivosť doručenia nie je garantovaná, pretože sa nemusí vždy podariť doručiť paket v správnom poradí alebo bez straty.

Fragmentácia a zrekonštruovanie: ak je veľkosť paketu väčšia ako maximálna veľkosť prenosnej jednotky (MTU) siete, môže byť rozdelený na menšie fragmenty, ktoré očísľuje a odošle. Na strane prijímateľa sú tieto fragmenty zase zrekonštruované do pôvodného paketu.

Nezávislosť na médiu: IP je nezávislý na médiu, čo znamená, že môže byť použitý na prenos dát cez rôzne typy prenosových médií, ako sú napr. drôty, optické vlákna, bezdrôtové spojenie atd.

Routovanie: IP umožňuje routovanie dát v sieti, čo znamená, že pakety sú posielané cez rôzne zariadenia (smerovače), aby sa dostali z počiatočnej stanice k cieľovej stanici. Tento proces sa uskutočňuje na základe IP adres a smerovacej tabuľky.

Podpora rôznych služieb: IP môže byť použitý pre rôzne typy komunikácie vrátane:

- unicastu (komunikácia medzi jedným odosielateľom a jedným prijímateľom),
- multicastu (komunikácia medzi jedným odosielateľom a viacerými prijímateľmi),
- broadcastu (komunikácia z jedného odosielateľa na všetkých prijímateľov v rovnakej sieti)

b3 Porovnajte hlavičky paketu IPv4 a IPv6, použite obrazovú prílohu č. 8

IPv4 paket sa skladá zo 14 polí a má 20 bytovú hlavičku.

Version – označuje typ paketu (IPv4 alebo IPv6).

DS (Differentiated Services or DiffServ) – predtým známy ako typ služby (ToS) pole, DS pole sa používa na určenie priority každého paketu.

Time to Live (TTL) – slúži na nastavenie limitu životnosti paketu, jeho hodnota je nastavená odosielateľom (napr. router) a hodnota klesá vždy keď ho spracuje iný router. Keď sa hodnota rovná nule, router paket zahodí a pošle ICMP (Internet Control Message Protocol) správu zdrojovej IP adresy (odosielateľovi). Pretože router znižuje hodnotu TTL každého paketu, musí taktiež prepočítať hodnotu v poli **Header Checksum**.

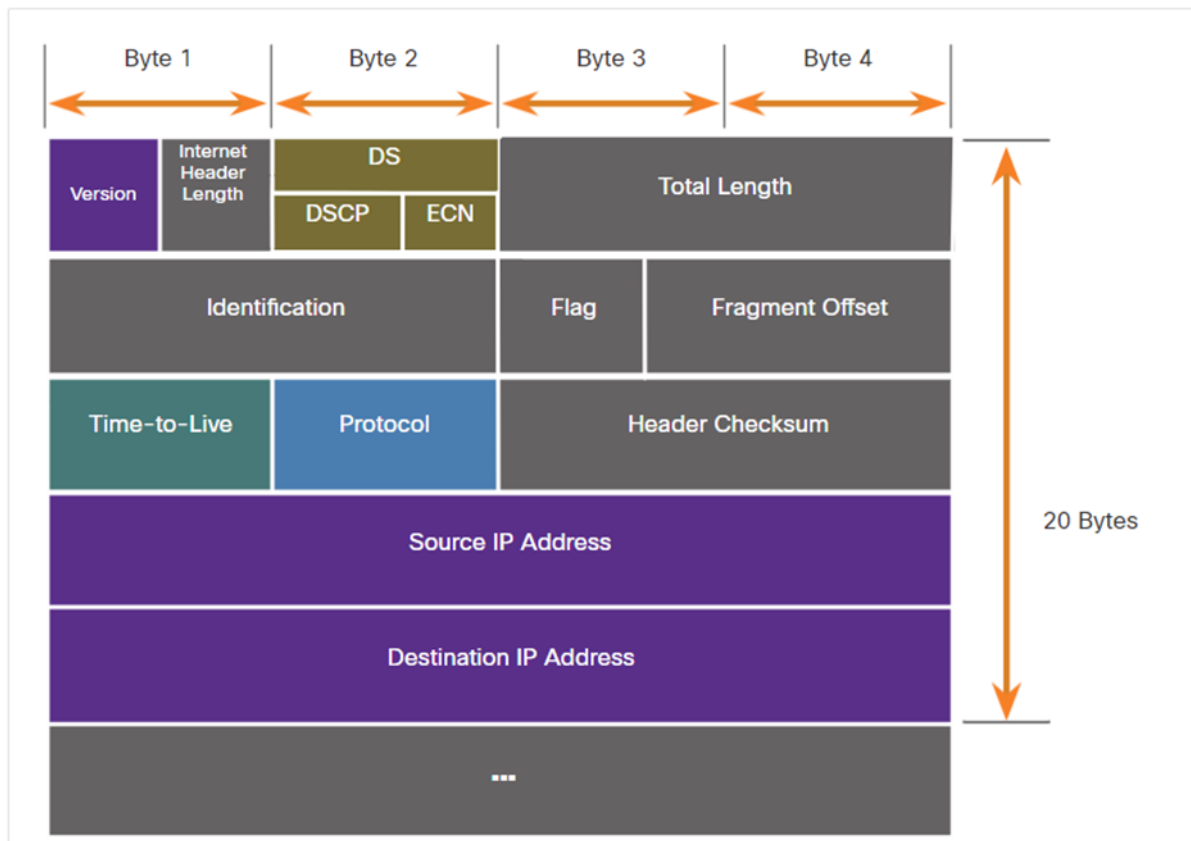
Protocol – používa sa na identifikáciu typu transportného protokolu v ktorom sú uložené samotné dáta (payload), ktoré paket nesie/obsahuje. Transportné protokoly: ICMP hodnota (1), TCP hodnota (6), UDP hodnota (17).

Header Checksum – používa sa k odhaleniu korupcie paketu.

Source IPv4 Address – reprezentuje zdrojovú IPv4 adresu paketu, táto adresa je vždy unicast adresa.

Destination IPv4 Address – reprezentuje cieľovú IPv4 adresu paketu, táto adresa môže byť unicast, multicast alebo broadcast adresa

Fields in the IPv4 Packet Header



IPv6 paket sa skladá zo 8 polí a má 40 bytovú hlavičku.

Version – označuje typ paketu (IPv4 alebo IPv6).

Traffic Class – je ekvivalentné/rovnaké ako služba DS.

Flow Label – uľahčuje prácu router-om. Všetky pakety s rovnakým flow label, dostanú rovnaký typ spracovania router-om.

Payload Length – označuje dĺžku dátovej časti alebo náklad v IPv6 pakete.

Next Header – je ekvivalentné/rovnaké ako pole Protocol

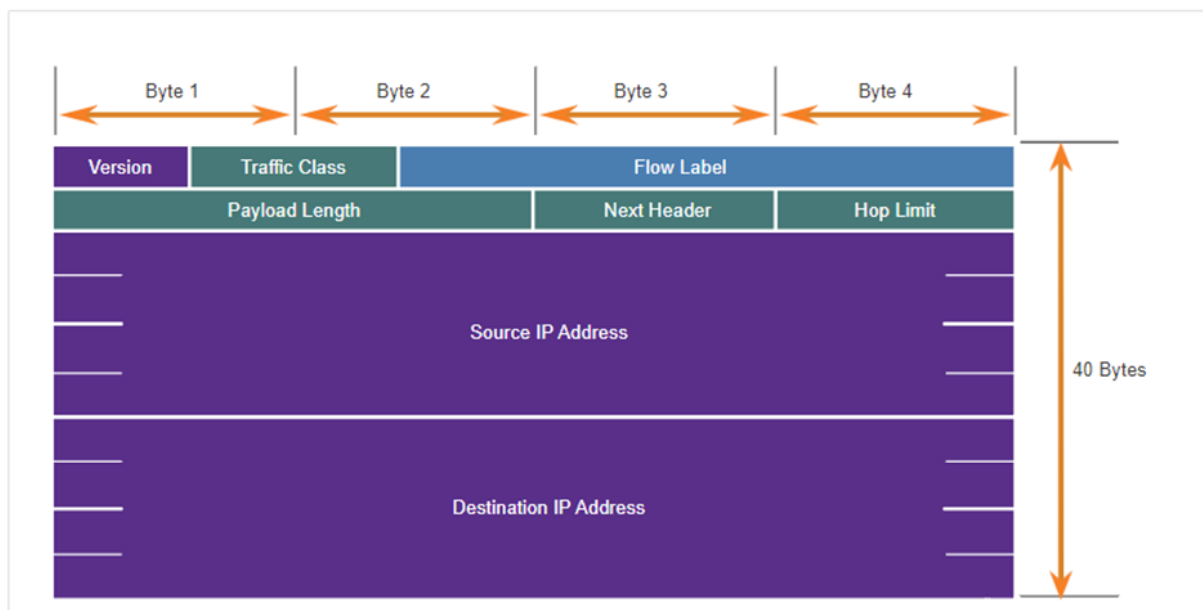
Hop Limit – nahrádza IPv4 TTL pole. Hodnota sa znižuje po jednej vždy keď router odošle paket. Keď počítadlo dosiahne nulu, paket sa zahodí a odošle ICMPv6 správu odosielaťúcemu hosťovi, že sa paket nedostal do cieľa pretože sa vyčerpal Hop Limit.

Info. navyše ku Hop Limit sekcii – IPv6 nepotrebuje pole Checksum pretože sa táto funkcia vykonáva na oboch nižších aj vyšších vrstvách. To znamená, že hodnota v poli Checksum nemusí byť vždy prepočítaná router-om keď sa znižuje hodnota v Hop Limit poli a to zlepšuje výkon siete.

Source IPv6 Address – označuje IPv6 adresu odosielaťcu.

Destination IPv6 Address – označuje IPv6 adresu prijímateľa.

Fields in the IPv6 Packet Header



c1 Popíšte rôzne formy platobného styku (elektronické bankovníctvo)

Samoobslužná zóna

Medzi prvé samoobslužné zariadenia v bankovníctve na svete patria bankomaty na výber hotovosti prostredníctvom magnetických platobných kariet. Základom práce s takýmito zariadeniami je platobná karta k účtu. Platobná karta je považovaná za bezpečný platobný nástroj, jej použitie je viazané na bezpečnostné PIN číslo. Z uvedených dôvodov je platobná karta používaná ako bezpečný identifikátor klienta aj v ďalších samoobslužných zariadeniach, prostredníctvom ktorých je možné dozvedieť sa stav svojho účtu, ale aj vykonávať platobné príkazy.

Technické zariadenia v samoobslužnej zóne

Samoobslužná zóna je spravidla vybavená týmito zariadeniami:

- bankomat kombinovaný s nočným trezorom
- bankomat kombinovaný so zmenárenským automatom a depozitom
- informačný terminál
- transakčný terminál
- autosafe (automatické bezpečnostné schránky)

Samoobslužná zóna je klientom prístupná 24 hodín denne.

Homebanking

Homebanking patrí medzi najčastejšie používanú telebankingovú službu. Okrem výhod pre klienta (úspora času, celodenný kontakt s bankou, atď.) odbremenil homebanking aj zamestnancov banky tým, že istú čas svojej práce preniesli na klientov a mohli sa venovať inému zákazníkovi, čím sa zvýšila efektívnosť.

Každá banka má aj svoj vlastný softvér, ku ktorému klient dostane svoj prístupový kód, ktorý môže byť buď alfanumerickou kombináciou alebo nejaké identifikačné zariadenie. Po spojení sa s bankou a identifikovaní umožní systém klientovi uskutočňovať aktívne a pasívne operácie v rámci vlastného účtu.

Homebanking je súčasťou informačného systému banky, je na ňom závislý. Väčšinou funguje spôsobom off-line vo vzťahu k centrálnemu informačnému systému banky, čiže jednotlivé finančné operácie sa realizujú po dávkach.

V rámci homebankingu má klient možnosť vykonávať 2 základné typy operácií:

- pasívne - informácie o zostatku na účte, obratoch za určité obdobie, prevodoch, kurzových lístkoch a pod.
- aktívne - prevody medzi účtami, t. j. domáci a zahraničný platobný styk, inkasá, zasielanie textových správ a pod.

Internet Banking

Rozvoj celosvetovej počítačovej siete prináša do vzťahu medzi klientom a bankou novú kvalitu a komfort. Internet banking je vlastne istá obdoba homebankingu, avšak na kvalitatívne vyššej úrovni, pretože pri použití odpadá potreba špeciálneho softvéru, ktorý zabezpečoval komunikáciu s bankou, a stačí obyčajný internetový prehliadač (Internet Explorer alebo Netscape Navigator). Avšak prvej návšteve banky sa klient ani tu nevyhne, pretože aj o túto službu musí požiadať banku, ktorá mu poskytne identifikačný kód. Po otvorení webovej stránky v danej banke stačí, ak si klient vyberie voľbu internet banking a po správnej identifikácii má možnosť uskutočňovať pasívne, resp. aj aktívne operácie so svojim účtom v závislosti od jednotlivých podmienok každej banky.

Princíp internet bankingu je analogický ako pri homebankingu. Na prekonanie rizika verejnej siete sa používajú rôzne druhy kryptovania a šifrovania dát. Jedným z možných riešení je elektronický kľúč, ktorý v spojení so softvérom ponúka nasledovné bezpečnostné funkcie:

- autentizácia klienta
- autentizácia servera
- šifrovanie komunikácie
- generovanie certifikačných dát

Spektrum služieb ponúkaných prostredníctvom internet bankingu má narastajúcu podobu. Vo vyspelých krajinách ponúkajú banky prostredníctvom internet bankingu aj pôžičky a hypotekárne úvery, obchodovanie na burze, poistenie a mnohé ďalšie služby.

Mailbanking

V rámci mailovej bankovej služby sa poskytujú klientele promptné informácie o operáciách na účte v elektronickej podobe. Klient môže získať prehľad obrátov na účte, resp. e-mailový výpis, t.j. elektronický opis štandardného výpisu s periodicitou zadanou pri papierovej forme.

Telefónbanking

Systémy telefónbankingu ako distribučné kanály bankových produktov nemajú z hľadiska jednoduchosti prístupu ku službám konkurenciu. Vznik prvých takýchto systémov umožnilo v osemdesiatych rokoch napredovanie a rozvoj telefónnych a telekomunikačných systémov. V súčasnosti je telefónbanking najdostupnejšou telebankingovou službou. Je určený pre fyzické osoby, ktorým umožňuje operatívne sa skontaktovať so svojou bankou v priebehu celých 24 hodín. Namiesto počítača sa na spojenie klienta s bankou používa telefón.

Ako funguje telefónbanking?

Zavolaním na verejné telefónne číslo príslušnej banky je možné sa spojiť s hlasovým informačným systémom banky, ktorý sám navádza klienta, ako sa pohybovať v menu. Jediné, čo je na to potrebné, je telefónny aparát s tónovou voľbou. V prípade zasielania príkazov na úhradu je nutné si zakúpiť elektronický osobný kľúč. V zmluve o využívaní telebankingových služieb sa zvolia účty, ktoré sa budú cez komunikačný kanál využívať a spôsob identifikácie podľa toho, aký rozsah operácií je v záujme realizovania.

GSM banking

Ďalšou formou elektronického bankovníctva je GSM banking. V situácii, keď Slovensko zažíva obrovskú explóziu nárastu počtu užívateľov mobilných telefónov, bolo zavedenie tejto služby nevyhnutnosťou. Niektoré banky s touto formou komunikácie začali len prednedávnom, a niektoré sa zasa na zavedenie tohto produktu do svojho produktového portfólia ešte len pripravujú. Na to, aby klient mohol využívať služby GSM bankingu, potrebuje mať podpísanú dohodu s bankou, ktorá mu buď za poplatok alebo zdarma (v závislosti od podmienok každej banky) poskytne telefonický kontakt na seba, mobilný telefón so sprístupnenou službou SMS. S bankou potom klient komunikuje na základe SMS správ, čiže odošle do banky požiadavku a ona sa mu okamžite vráti takisto vo forme SMS správy. Je to vhodný spôsob, ako aktuálne a priebežne dostávať informácie o stave na účte alebo ako získať aktuálne kurzy zahraničných mien. Okrem pasívnych operácií niektoré banky sprístupnili cez mobilnú sieť aj aktívne operácie a účtom klienta, ktorých realizácia je však už komplikovanejšia.

Mobilbanking

Umožňuje manažérom alebo klientom s menším počtom bankových operácií pripravovať a posilať príkazy do banky, získať informácie o aktuálnom zostatku na účte, o posledných obrátových položkách, kurzový lístok a pod. Namiesto PC je založený na využívaní vreckového počítača Palm Pilot. Má určitú podobnosť s homebankingom, ktorý využíva PC a SMS bankingom, využívajúcim mobilný telefón.

c2 Vysvetlite spôsob vykonania príkazu priradenia, uveďte príklad

Operátor priradenia, teda v programovaní = slúži na to, aby sme napríklad priradili hodnotu jednej premennej druhej premennej. Teda napríklad máme:

```
int cislo1 = 2;
```

```
int cislo2 = 8;
```

```
cislo1 = cislo2;
```

Hodnotu premennej z čísla 1 som priradil premennej číslo 2. Celé sa to volá priraďovací príkaz.

9. Sieťové – Siete LAN, WAN, VLAN – Filip

a)

1. Definujte siete PAN, LAN, VLAN, MAN, WAN
2. Načrtnite niektoré topológie sietí LAN

3. Porovnajte siete typu client - server a sietí peer – to – peer
4. Popíšte jednotlivé komponenty sietí
5. Vysvetlite intranet, extranet a internet

b)

1. Načrtnite zariadenia, ktoré používame v LAN a WAN
2. Porovnajte konvergované siete a tradičné oddelené siete, použite obrazovú prílohu č. 9
3. Navrhните najvhodnejšie riešenie na vybudovanie WLAN v domácnosti alebo v kancelárii

c)

1. Vytvorte cenovú ponuku vami navrhnuťej siete WLAN a uveďte základné metódy tvorby ceny v podniku
2. Vymenujte aritmetické operátory a vysvetlite ich význam a spôsob použitia

a1 Definujte siete PAN, LAN, VLAN, MAN, WAN

PAN (Personal area network) – osobná počítačová sieť, ktorá je spravidla tvorená počítačmi umiestnenými v tesnej blízkosti alebo počítačom a iným elektronickým zariadením (tlačiareň, PDA, mobilný telefón).

Na prenos údajov najčastejšie používa bezdrôtové pripojenie (napr. *bluetooth*) a jej primárnym cieľom je prenos a synchronizácia údajov. Prenosová rýchlosť dosahuje zvyčajne niekoľko desiatok Mb/s.

LAN (Local area network) – lokálna počítačová sieť, ktorá pracuje v režime neustáleho spojenia a na komunikáciu medzi počítačmi nepotrebuje nadväzovať spojenie (nepoužíva prostriedky pre diaľkový prenos údajov).

Maximálne vzdialenosti medzi počítačmi sú stovky metrov až niekoľko kilometrov. Väčšinou sú umiestnené v jednej alebo viacerých blízkych budovách. Rýchlosť tohto typu sietí dosahuje rádovo Gb/s.

Virtuálna LAN je obdobou klasickej lokálnej siete s tým, že LAN závisí na *fyzickom usporiadaní* a prepojení, zatiaľčo VLAN vzniká *logicky* vnútri fyzickej LAN.

MAN (Metropolitan area network) – mestská (metropolitná) sieť, špeciálny typ LAN, pre ktorú je charakteristická väčšia rozloha ako pre LAN v pôvodnom slova zmysle.

Zatiaľ čo klasická LAN je štandardne využívaná zamestnancami vlastníka, MAN slúži širokému spektru používateľov, ktorí za umožnenie prístupu platia prevádzkovateľovi.

WAN (Wide area network) – rozľahlá sieť, ktorá používa prostriedky pre diaľkový prenos údajov a nevyžaduje neustále spojenie. Jej rozloha je v podstate neobmedzená.

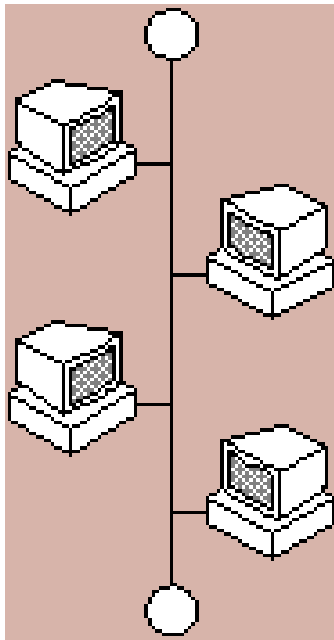
Môže spájať počítače na väčších územiach (ako štát, kontinent) alebo aj na celom svete. Základ WAN siete je tvorený hlavným komunikačným kanálom označovaným ako chrbtica (*backbone*), pre ktorú je typická vysoká prenosová rýchlosť získaná vďaka optickým vláknam. K serverom umiestneným na chrbticovej sieti sa pripájajú ďalšie počítače alebo počítačové siete prostredníctvom vytáčaného alebo trvalého pripojenia. Rýchlosť v rámci WAN závisí od typu spojenia a pohybuje sa od niekoľkých kb/s do desiatok Gb/s. Typickými predstaviteľmi WAN sú siete ISDN, DSL a 3G. Najznámejšou WAN je Internet.

a2 Načrtnite niektoré topológie sietí LAN

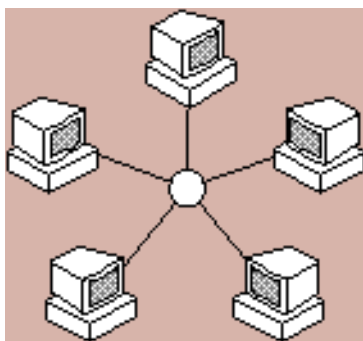
Kľúčovú úlohu v sieťach akéhokoľvek typu zohrávajú aktívne prvky. Ich úlohou je prepájať jednotlivé časti sietí, menia typy rozhraní, spôsoby komunikácie, zaisťujú bezpečnosť a riadia sieť. Sieť môže byť prepojená priamo z počítača do počítača, alebo s využitím týchto aktívnych prvkov.

Všetky návrhy siete vychádzajú z troch základných topológií:

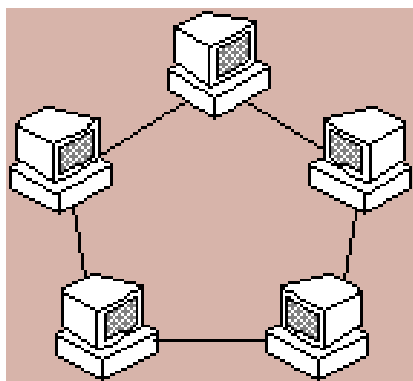
» **Zbernica** - kostrou siete je spojovacie vedenie, ku ktorému sú pripojené jednotlivé uzly siete pomocou príslušných odbočovacích prvkov bez centrálnej alebo riadiacej stanice. Dátové správy sa šíria vedením všetkými smermi a všetky stanice k nim majú prístup. Na spojenie počítačov sa využíva najčastejšie koaxiálny kábel s prenosovou rýchlosťou 10 Mbit/s. Je najjednoduchšia, možno ju ľahko rozširovať a upravovať. Pri poruche jedného z uzlov neprichádza k výpadku prenosu údajov v sieti. Koncesegmentov musia byť ukončené zakončovacím odporom - terminátorom.



» **Hviezda** - v centre siete pôsobí centrálny uzol, ku ktorému sú pripojené všetky ostatné uzly siete najčastejšie pomocou symetrického kábla. Centrálny uzol je buď pasívny koncentrátor (HUB - distribuuje signál vysielaný stanicami) alebo aktívny opakovač či rozbočovač (SWITCH). Dátové správy sa šíria vedením a takisto všetky stanice k nim majú prístup. Táto topológia umožňuje tiež jednoduché a lacné rozširovanie, avšak pri poruche centrálného uzla je celá sieť vyradená z prevádzky.



» **Kruh** - stanice sú prepojené vedením do tvaru súvislého kruhu. Dátové správy sa odovzdávajú postupne jedným smerom medzi stanicami. Pri poruche jedného z uzlov príde k výpadku sieťového prenosu. Inkriminovaný prvok siete je potrebné premostiť.

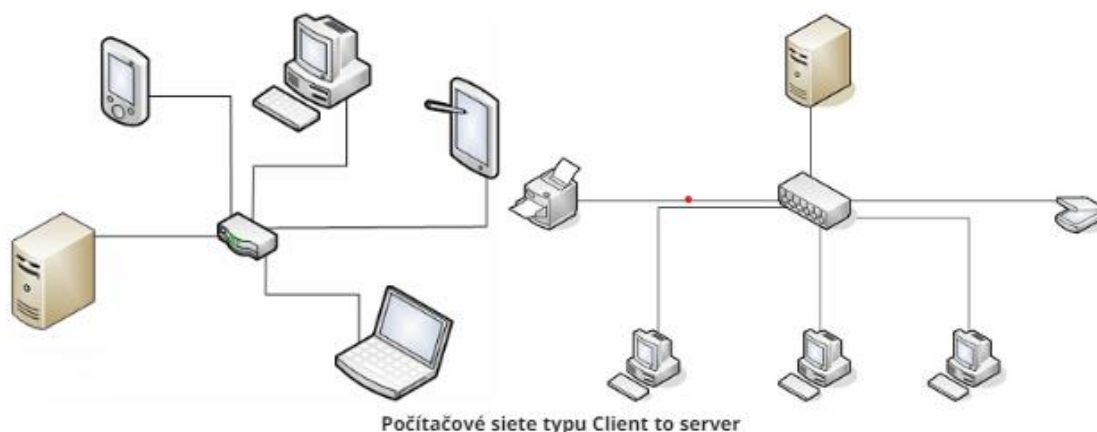


a3 Porovnajete siete typu client - server a siete peer – to – peer

Klient-server je model, v ktorom existuje jeden alebo viac počítačov, ktoré plnia úlohu servera. Ďalšie počítače potom plnia rolu klientov. Klienti sa pripájajú k serveru a požadujú dáta alebo služby. Server potom klientom poskytuje požadované informácie alebo služby. Tento model sa často používa v organizáciách alebo inštitúciách, kde je potrebná silná centralizácia a kontrola prístupu k dátam.

Výhody: ľahká správa siete, nie je nutné kupovať server ani žiadne sieťové operačné systémy (peer – to – peer je obsiahnutá vo Windows)

Nevýhody: pri veľkom počte počítačov je ťažké udržať prehľad o dátach, dáta sú len málo chránené proti zneužitiu, konfigurácia prístupových práv je neúčinná.

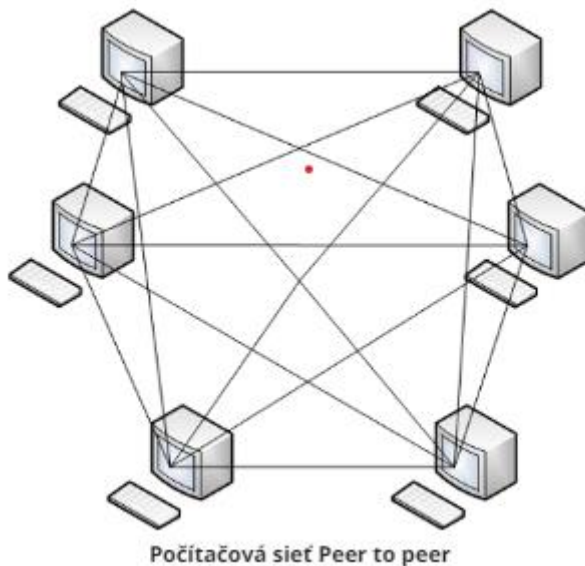


Peer-to-Peer (P2P) je model, v ktorom každý počítač (uzol) v sieti má rovnakú rolu a môže byť ako klientom, tak aj serverom. V tejto architektúre nie je žiadny počítač, ktorý by plnil funkciu hlavného uzla alebo centrálného servera. Dáta sú zdieľané medzi

uzlami priamo, bez potreby sprostredkovateľa. P2P siete sú často využívané na zdieľanie súborov, hudby alebo videa.

Výhody: vysoká bezpečnosť dát, prehľadnosť, jednoduchá konfigurácia

Nevýhody: náklady na nákup serveru, potreba vysokokvalifikovaného pracovníka (administrátora), potreba zabezpečiť sieťový operačný systém.



a4 Popíšte jednotlivé komponenty sietí

Hardverové komponenty - zahrňuje všetky technické prostriedky siete. Patria sem aj prostriedky, ktorými je realizované vlastné prepojenie siete.

Počítače: Jadro každej počítačovej siete. Môžu to byť stolné počítače, prenosné počítače, smartphony, tablety alebo akékoľvek iné zariadenia, ktoré sa dokážu pripojiť k sieti.

Servery: Výkonné počítače, ktoré ukladajú a spracovávajú dáta pre klientske počítače. Môžu to byť napríklad webové servery, databázové servery, e-mailové servery a iné.

Smerovač (router) je zariadenie, ktorého úlohou je usmerňovať tok paketov medzi podsietami tak, aby sa každý paket dostal k svojmu adresátovi. Každý smerovač v sieti má prehľad o tom, aké podsiete sú k nemu pripojené a ktoré ďalšie smerovače sa v nich nachádzajú. S týmito smerovačmi si vzájomne tieto informácie vymieňajú.

Rozbočovač (Hub)

Koncentruje pripojenia, inými slovami zoskupuje počítače a umožňuje sieti vidieť ich ako jednoduchú jednotku. Rozbočovač sa tiež používa keď chceme prepojiť viac ako dva počítače

– to je rovnaká funkcia ako u prepínača. Rozbočovač, na rozdiel od prepínača, nič neoptimalizuje a len ako rozbočka posiela komunikáciu z jednej prípojky na tie ostatné.

Most (Bridge) je inteligentný prvok, ktorý sa zaujíma o prenášané dáta. Plní dve funkcie:

1. filtrácia paketov – most si prečíta cieľovú adresu paketu a prepustí ho len do tej časti siete, v ktorej je cieľ paketu. Filtrovaním sa podstatne znižuje zaťaženie siete.
2. prepojenie dvoch sietí rôznych štandardov. Pracuje v linkovej vrstve ISO/OSI takže fyzické odlišnosti siete ho neovplyvňujú.

Most je veľmi často integrovaný do hubov (ktorým sa potom vraví L2 switch), často úlohu mostov plní softvér sieťového operačného systému ktorý pakety filtruje medzi viacerými sieťovými kartami.

PREPÍNAČ (SWITCH)

je funkčne podobný mostu, označuje sa aj ako viacportový most. Switch posiela údaje len na ten port, pre ktorý sú určené, rieši filtráciu paketov prebiehajúcich medzi jednotlivými portami. Prichádzajúce pakety filtruje na základe MAC adresy a posiela ich ďalej, prepúšťa pakety iba medzi vysielajúcou a prijímajúcou stanicou – prepína pakety medzi dvoma portami.

BRÁNA (GATEWAY)

je najzložitejším aktívnym prepojovacím prvkom siete. Hlavnou funkciou je pripojovania sietí LAN na úplne odlišné sieťové architektúry, zabezpečenie prevodu medzi odlišnými sieťovými koncepciami.

Opakovač (repeater)

Je najjednoduchším aktívnym prvkom kabeláže, pretože iba zosilňuje (opakuje) ním prechádzajúci signál. Je to zariadenie s dvoma rovnakými konektormi. Používa sa tam, kde je kábel taký dlhý, že na jeho konci by už nebol dostatočne silný signál.

PREVODNÍK (TRANSCEIVER)

je podobný opakovaču. Funkcie sú obdobné, okrem zosilnenia signálu umožňuje prechod z metaliky na optiku a opačne.

Modem je zariadenie, ktoré umožňuje pripojenie siete k internetu prostredníctvom komunikačnej linky, ako je telefónna linka, káblová linka, optický kábel atď.

Káble:

- Káble spájajú zariadenia v sieti a umožňujú prenos dát medzi nimi. Existuje viacero typov káblov, z ktorých najbežnejšie sú:
 - **Ethernetové káble:** Používajú sa na prenos dát v lokálnych sieťach (LAN).
 - **Optické káble:** Používajú sa na prenos dát na dlhé vzdialenosti.

- **Koaxiálne káble:** Používajú sa v starších sieťach a dnes sa už bežne nepoužívajú.

Softverové komponenty - programové vybavenie, ktoré v spolupráci s hardware-om siete zabezpečuje funkcie siete. U niektorých operačných systémov sú tieto funkcie už jeho súčasťou. Väčšinou je realizované pomocou rezidentných programov.

Operačné systémy:

- Operačné systémy, ako napríklad Windows, macOS a Linux, obsahujú zabudovaný sieťový softvér, ktorý umožňuje počítačom pripojiť sa k sieti a komunikovať s ostatnými zariadeniami.
- Poskytujú základné sieťové funkcie, ako napríklad správu sieťových rozhraní, smerovanie IP paketov a správu sieťových pripojení.

Sieťové protokoly:

- Sieťové protokoly definujú pravidlá pre komunikáciu medzi zariadeniami v sieti.
- Určujú formát paketov dát, spôsob ich prenosu a spôsob interpretácie prijatých dát.
- Bežné sieťové protokoly zahŕňajú:
 - **TCP/IP:** Súbor protokolov, ktorý tvorí základ internetu a používa sa v rôznych sieťach.
 - **HTTP:** Protokol používaný na prenos webových stránok z webových serverov do prehliadačov.
 - **FTP:** Protokol používaný na prenos súborov medzi počítačmi.
 - **SMTP:** Protokol používaný na odosielanie e-mailov.
 - **POP3:** Protokol používaný na prijímanie e-mailov.

Sieťové aplikácie:

- Sieťové aplikácie sú softvérové programy, ktoré umožňujú používateľom pristupovať k sieťovým zdrojom a službám.
- Bežné sieťové aplikácie zahŕňajú:
 - **Webové prehliadače:** Umožňujú používateľom pristupovať k webovým stránkam na internete.
 - **E-mailové klienti:** Umožňujú používateľom odosielať a prijímať e-maily.
 - **Klienti na instant messaging:** Umožňujú používateľom komunikovať s ostatnými používateľmi v reálnom čase.
 - **Sieťové hry:** Umožňujú hráčom hrať hry online s ostatnými hráčmi.

a5 Vysvetlite intranet, extranet a internet

Internet je globálna sieť prepojených počítačových sietí, počítačov a iných zariadení. Umožňuje výmenu informácií, komunikáciu a zdieľanie zdrojov ako sú multimédiá a dáta.

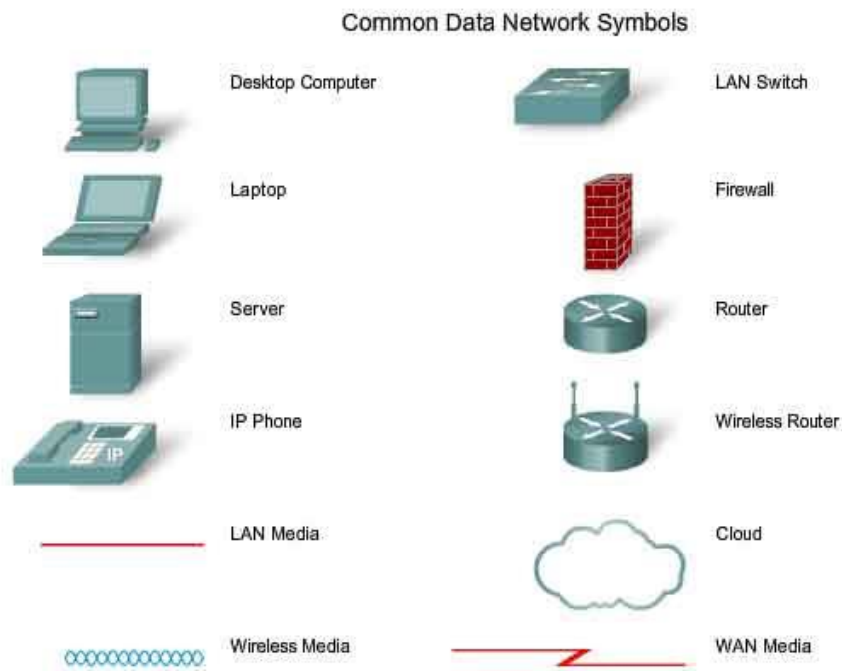
Extranet je rozšírenie internetu, ktoré umožňuje vybraným externým používateľom prístup k interným informáciám a systémom spoločnosti. Je to bezpečný a riadený priestor, kde môžu externí partneri, zákazníci alebo dodávatelia využívať určité informácie a služby bez toho, aby mali prístup k interným systémom spoločnosti. Extranet funguje na princípe autorizácie a prístupových práv. K určitým informáciám a systémom teda majú prístup len určení užívatelia. Tieto informácie môžu zahŕňať napríklad objednávky, faktúry, stavy zásob a ďalšie.

Intranet je vnútorná sieť spoločnosti, ktorá slúži ako platforma na zdieľanie informácií, komunikáciu a spoluprácu v rámci organizácie. Určený iba pre interné použitie (firmy, organizácie, inštitúcie) a prístup k nemu majú iba overení a identifikovaní užívatelia (typicky zamestnanci).

b1 Načrtnite zariadenia, ktoré používame v LAN a WAN

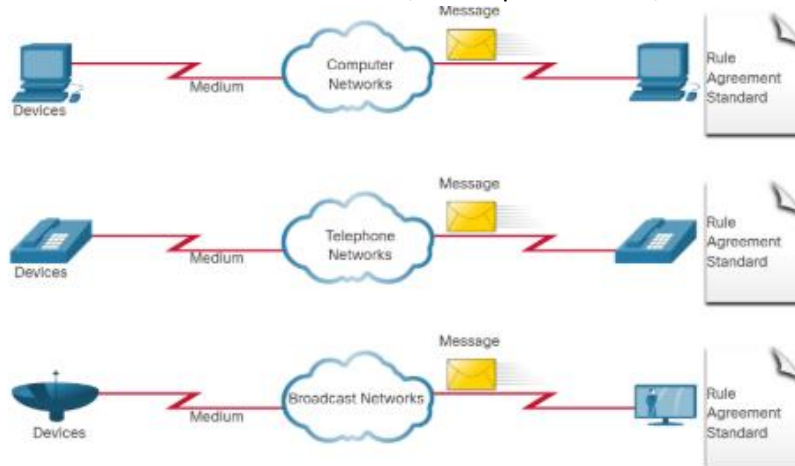
Hlavnými komponentmi siete LAN sú počítače, sieťové zariadenia, káble, prepínače, smerovače a server.

Hlavnými komponentami WAN sú routery, multiplexery, optické káble, satelitné systémy či modémy.



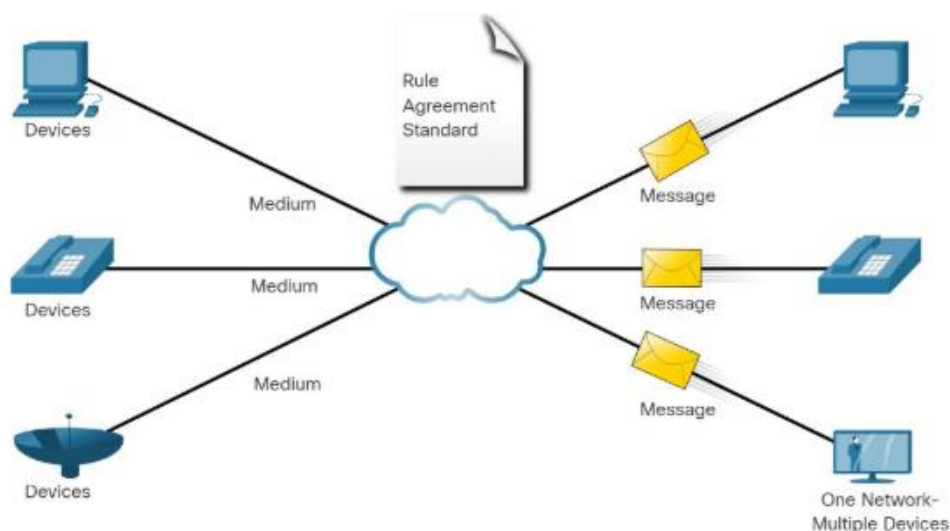
b2 Porovnajte konvergované siete a tradičné oddelené siete, použite obrazovú prílohu č. 9

Tradičná rozdelená sieť je typ siete, v ktorej sú rôzne funkcie siete, ako napríklad prenos dát, hlasová komunikácia a video, rozdelené do samostatných sietí. To znamená, že pre každú funkciu siete je potrebná samostatná infraštruktúra, ako napríklad káble, smerovače a prepínače.



Obrázok 1

Konvergovaná sieť je naproti tomu typ siete, v ktorej sú všetky funkcie siete prenášané cez jednu infraštruktúru. To znamená, že dáta, hlas a video sa prenášajú cez rovnaké káble, smerovače a prepínače.



Vlastnosť	Tradičná rozdelená sieť	Konvergovaná sieť
Infraštruktúra	Samostatná pre každú funkciu siete	Jedna pre všetky funkcie siete
Náklady	Vysoké, pretože je potrebná viacnásobná infraštruktúra	Nížšie, pretože je potrebná len jedna infraštruktúra
Flexibilita	Nízka, pretože je ťažké meniť sieť	Vysoká, pretože je jednoduché meniť sieť
Škálovateľnosť	Nízka, pretože je ťažké pridávať nové zariadenia	Vysoká, pretože je jednoduché pridávať nové zariadenia
Spoľahlivosť	Nízka, pretože porucha jednej siete môže ovplyvniť všetky funkcie	Vysoká, pretože porucha jednej siete ovplyvní len jednu funkciu

b3 Navrhňte najvhodnejšie riešenie na vybudovanie WLAN v domácnosti alebo v kancelárii

Veľkosť a rozloženie priestoru:

- **Malá domácnosť/kancelária:** Vhodný je jeden bezdrôtový router so štandardom Wi-Fi 6 alebo novším, ktorý dokáže pokryť signálom celú oblasť.

- **Väčšia domácnosť/kancelária:** V tomto prípade budete potrebovať viacero bezdrôtových routerov alebo prístupových bodov (AP), ktoré sa navzájom prekrývajú a vytvárajú tak bezproblémovú sieť Wi-Fi.
- **Viacero poschodí:** Na zaistenie silného signálu na všetkých poschodiach budete potrebovať bezdrôtové routery/AP na každom poschodí.

Počet zariadení:

- **Malý počet zariadení:** Ak máte len pár zariadení, ktoré sa pripájajú k Wi-Fi, postačí vám bežný bezdrôtový router.
- **Veľa zariadení:** Ak máte veľa zariadení, ktoré sa pripájajú k Wi-Fi, budete potrebovať bezdrôtový router s technológiou MU-MIMO (Multiple-User, Multiple-Input, Multiple-Output), ktorá umožňuje routeru komunikovať s viacerými zariadeniami súčasne.

Potrebná rýchlosť internetu:

- **Základné použitie (prehliadanie webu, e-mail):** Vhodný je bezdrôtový router s rýchlosťou Wi-Fi 5 GHz.
- **Streamovanie videa, hranie online hier:** Vhodný je bezdrôtový router s rýchlosťou Wi-Fi 6 alebo novším.

Ďalšie faktory:

- **Steny a prekážky:** Ak máte v dome/kancelárii hrubé steny alebo iné prekážky, ktoré môžu blokovať signál Wi-Fi, budete potrebovať bezdrôtový router s vyšším výkonom alebo systém Wi-Fi mesh.
- **Rozpočet:** Bezdrôtové routery sú dostupné v rôznych cenových reláciách. Pri výbere routera zvážte svoje potreby a rozpočet.

c1 Vytvorte cenovú ponuku vami navrhnuitej siete WLAN a uveďte základné metódy tvorby ceny v podniku

Text

c2 Vymenujte aritmetické operátory a vysvetlite ich význam a spôsob použitia

Operátor	Účel	Príklad
----------	------	---------

+	Súčet dvoch čísel.	[Medzisúčet]+[Daň z obratu]
-	Zistenie rozdielu medzi dvoma číslami alebo určenie zápornej hodnoty čísla.	[Cena]-[Zľava]
*	Vynásobenie dvoch čísel.	[Množstvo]*[Cena]
/	Vydelenie prvého čísla druhým číslom.	[Celkový súčet]/[Počet položiek]
\	Zaokrúhlenie oboch čísel na celé čísla, vydelenie prvého čísla druhým číslom a potom skrátenie výsledku na celé číslo.	[Registrované]\[Miestnosti]
Mod	Vydelenie prvého čísla druhým číslom a potom vrátenie zvyšku.	[Registrované] Mod [Miestnosti]

^

Umocnenie čísla na exponent.

Číslo ^ Exponent

10. Sieťové – Hardvér počítačových sietí - Matúš

a)

1. Popíšte typy sieťových adaptérov (sieťových kariet)
2. Definujte priradenie sieťového adaptéra ku konkrétnej vrstve RM OSI
3. Vysvetlite štruktúru MAC adresy
4. Kategorizujte rozhranie sieťových adaptérov pre metalické médiá
5. Zdôvodnite potrebu ARP protokolu

b)

1. Definujte normu 568B a 568A
2. Popíšte rámec Ethernet
3. Vysvetlite, ako switch napĺňa tabuľku MAC adres, použite obrazovú prílohu č. 10

c)

1. Zaraďte počítač do majetku podniku a popíšte formy jeho opotrebenia a odpisovania v podniku
2. Vymenujte relačné a logické operátory a vysvetlite ich význam a spôsob použitia

a1 Popíšte typy sieťových adaptérov (sieťových kariet)

Existuje niekoľko rôznych typov sieťových adaptérov alebo sieťových kariet, ktoré sa používajú na pripojenie počítačov a ďalších zariadení k lokálnej sieti.

Ethernetové karty (Ethernet adapters): Ethernet je najrozšírenejší typ lokálnej siete (LAN) a Ethernetové karty sú najbežnejším typom sieťového adaptéra. Tieto karty sa pripájajú do slotov PCI, PCIe alebo USB portov počítača a poskytujú pripojenie pomocou kábla Ethernet.

Wi-Fi adaptéry (Wi-Fi adapters): Wi-Fi adaptéry umožňujú pripojenie k bezdrôtovej sieti (Wi-Fi). Tieto adaptéry môžu byť buď vstavané do počítača alebo pripojené cez USB port. Poskytujú bezdrôtové pripojenie k sieti na základe štandardov Wi-Fi.

Bluetooth adaptéry (Bluetooth adapters): Bluetooth adaptéry umožňujú bezdrôtové pripojenie k ďalším zariadeniam vybaveným technológiou Bluetooth, ako sú klávesnice, myši, slúchadlá a ďalšie. Tieto adaptéry sa tiež môžu pripojiť k internetu pomocou mobilného zariadenia s podporou Bluetooth a zdieľať pripojenie cez osobný hotspot.

Modemové adaptéry (Modem adapters): Modemové adaptéry sa používajú na pripojenie k internetu prostredníctvom klasickej telefónnej linky (DSL), káblového internetu alebo iných typov prístupu k internetu. Tieto adaptéry môžu byť interné (vstavané do počítača) alebo externé (pripojené cez USB port).

Powerline adaptéry (Powerline adapters): Powerline adaptéry umožňujú pripojenie k sieti pomocou elektrických vedení v budove. Tieto adaptéry používajú existujúce elektrické káble na prenos dát a poskytujú alternatívu k bezdrôtovému alebo káblovému pripojeniu.

Fiber Optic adaptéry (Fiber Optic adapters): Fiber Optic adaptéry sú určené pre pripojenie k sieťam s optickými vláknami. Používajú sa pre vysokorýchlostné pripojenie a sú často využívané v podnikových a širokopásmových sieťach.

a2 Definujte priradenie sieťového adaptéra ku konkrétnej vrstve RM OSI

V sieťovom modeli OSI (Open Systems Interconnection), sieťový adaptér je zvyčajne priradený k fyzickej vrstve (Layer 1) a linkovej vrstve (Layer 2).

Fyzická vrstva (Layer 1): V tejto vrstve je sieťový adaptér zodpovedný za fyzickú časť prenosu dát. To zahŕňa transformáciu dát na elektrické signály (v prípade káblového pripojenia), optické signály (v prípade optického vlákna) alebo rádiové signály (v prípade bezdrôtového pripojenia). Sieťový adaptér sa stará o prenos týchto signálov medzi počítačom a fyzickým médiom, ako sú káble, optické vlákna alebo vzduch.

Linková vrstva (Layer 2): Linková vrstva je zodpovedná za riadenie priameho prenosu dát medzi dvoma zariadeniami v rámci jednej lokálnej siete (LAN). Sieťový adaptér v linkovej vrstve spracúva rámce, ktoré sú jednotkou údajov na tejto úrovni, a zabezpečuje ich prenos medzi zariadeniami v sieti. To zahŕňa detekciu a opravu chýb, riadenie toku dát a adresovanie pomocou MAC (Media Access Control) adries.

Sieťové adaptéry môžu zohrávať aj úlohu v nadstavbách nad týmito vrstvami, ako sú aplikačná vrstva (Layer 7) a sieťová vrstva (Layer 3), v závislosti od ich funkcií a použitia. Napríklad sieťový adaptér môže mať funkcie, ktoré umožňujú jeho identifikáciu v rámci siete pomocou IP adresy, čo súvisí s sieťovou vrstvou (Layer 3). Taktiež môže mať funkcie zamerané na šifrovanie alebo dešifrovanie dát, čo súvisí s vyššími vrstvami ako aplikačná vrstva (Layer 7).

a3 Vysvetlite štruktúru MAC adresy

MAC adresa (Media Access Control adresa) je unikátny identifikátor priradený sieťovému adaptéru na fyzickej alebo linkovej vrstve v sieťovom modeli OSI. Skladá sa zo 48-bitového čísla. Táto adresa sa používa na jednoznačnú identifikáciu zariadenia v lokálnej sieti (LAN). Každý sieťový adaptér (sieťová karta) má zaručenú jedinečnú MAC adresu. Jedinečnosť prvých troch bajtov (ID výrobcu) zaručuje IEEE, jedinečnosť posledných troch zabezpečujú jednotliví výrobcovia.

Štruktúra MAC adresy sa líši podľa typu sieťového adaptéra a použitého formátu, ale všeobecne pozostáva z nasledujúcich častí:

Prefix: Niektoré formáty MAC adries môžu obsahovať prefix, ktorý označuje výrobcu alebo typ zariadenia. Táto časť môže byť použitá na identifikáciu výrobcu adaptéra. V MAC adresách štandardu Ethernet (IEEE 802.3) napríklad prvé tri byty (24 bitov) označuje organizáciu priradenou kódom OUI (Organizationally Unique Identifier).

Identifikátor výrobcu (OUI): Identifikátor výrobcu je časť MAC adresy, ktorá jednoznačne identifikuje výrobcu alebo výrobcu sieťového adaptéra. Tieto hodnoty sú pridelené a spravované organizáciou IEEE (Institute of Electrical and Electronics Engineers).

Identifikátor adaptéra: Táto časť MAC adresy je jedinečná pre každý konkrétny adaptér vyrobený daným výrobcom. To znamená, že každý sieťový adaptér vytvorený v rámci danej organizácie má jedinečnú MAC adresu.

Štruktúra MAC adresy môže závisieť aj od použitého formátu, pričom najbežnejším formátom je zápis v šesťnástkovej sústave, kde každý bajt (8 bitov) je oddelený dvojbodkou alebo pomlčkou. Napríklad: 00:1A:2B:3C:4D:5E.

Je dôležité poznamenať, že MAC adresa je priradená sieťovému adaptéru výrobcu a zvyčajne sa mení len v prípade výmenných zariadení. MAC adresa sa používa na jednoznačnú identifikáciu zariadenia v rámci lokálnej siete a nezmení sa ani pri zmene IP adresy alebo presune zariadenia do inej siete. MAC adresa je spravidla 48-bitové číslo, ktoré sa kvôli prehľadnosti uvádza ako 12-miestne hexadecimálne číslo (napr. 08:00:69:02:01:FC, ako oddeľovače dvojíc sa používajú dvojbodky alebo pomlčky, prípadne sa číslo píše bez oddeľovania dvojíc číslíc).

a4 Kategorizujte rozhranie sieťových adaptérov pre metalické médiá

Sieťové adaptéry pre metalické médiá môžu byť kategorizované podľa rôznych kritérií, ako je napríklad rýchlosť prenosu dát, typ použitého pripojenia alebo použitý štandard.

Ethernetové adaptéry: sú najbežnejším typom sieťových adaptérov pre metalické médiá. Existujú rôzne varianty štandardu Ethernet, ktoré podporujú rôzne rýchlosti prenosu dát (napr. 10 Mbps, 100 Mbps, 1 Gbps, 10 Gbps a vyššie). Tieto adaptéry môžu byť pripojené pomocou káblových médií, ako sú koaxiálne káble (pre staršie varianty Ethernetu), neuzemnené a uzemnené dvojlinkové káble (Unshielded Twisted Pair - UTP) alebo stínované dvojlinkové káble (Shielded Twisted Pair - STP).

Powerline adaptéry: Powerline adaptéry umožňujú využívať existujúce elektrické vedenia v budove na prenos dát. Tieto adaptéry umožňujú vytvoriť domácu sieť cez elektrické vedenie a poskytujú alternatívu k bezdrôtovej alebo káblovej sieti.

DSL adaptéry (DSL adapters): DSL adaptéry sú určené pre pripojenie k internetu prostredníctvom digitálnej prístupovej linky (DSL). Tieto adaptéry sa používajú v prípade, že poskytovateľ internetových služieb ponúka pripojenie k internetu cez telefónnu linku.

Modemové adaptéry: Modemové adaptéry umožňujú pripojenie k internetu pomocou káblového internetu. Tieto adaptéry môžu byť pripojené k počítaču prostredníctvom káblového rozhrania, ako je napríklad Ethernet, USB alebo iné.

Tieto kategórie sieťových adaptérov pre metalické médiá poskytujú rôzne možnosti pripojenia k lokálnej sieti a umožňujú používateľom prispôsobiť svoje pripojenie podľa ich potrieb a prostredia.

a5 Zdôvodnite potrebu ARP protokolu

ARP (Address Resolution Protocol) je nevyhnutným protokolom v lokálnych sieťach, pretože umožňuje prekladanie IP adres na fyzické MAC adresy.

ARP pracuje na linkovej vrstve (Layer 2) v sieťovom modeli OSI (Open Systems Interconnection). Linková vrstva je druhá vrstva v OSI modeli a zodpovedá za riadenie priamej komunikácie medzi dvoma zariadeniami v rámci lokálnej siete (LAN)

Komunikácia v rámci lokálnej siete: Keď zariadenie potrebuje komunikovať s iným zariadením v tej istej lokálnej sieti, potrebuje znalosť jeho fyzickej MAC adresy. ARP umožňuje získanie tejto informácie, čím umožňuje priamu komunikáciu medzi zariadeniami.

Bezpečnosť a spoľahlivosť: ARP protokol pomáha zabezpečiť, že správy sú doručené správne zariadeniu. Bez ARP by zariadenia nemohli správne doručiť dáta na základe IP adresy, pretože každé zariadenie potrebuje vedieť, kde sa nachádzajú ostatné zariadenia v sieti.

Dynamické priradenie IP adresy: V súčasných sieťach sú IP adresy často pridelené dynamicky pomocou protokolov ako DHCP (Dynamic Host Configuration Protocol). Keď sa zmení IP adresa zariadenia, ARP umožňuje aktualizáciu priradenia IP adresy k fyzickým MAC adresám.

Údržba ARP tabuľky: Každé zariadenie v sieti udržiava ARP tabuľku, ktorá obsahuje mapovanie medzi IP adresami a fyzickými MAC adresami. Táto tabuľka sa pravidelne aktualizuje pomocou ARP odpovedí, čo umožňuje efektívnu komunikáciu v sieti.

ARP je špecifický pre linkovú vrstvu, pretože jeho hlavnou úlohou je mapovanie medzi IP adresami (ktoré sa nachádzajú na sieťovej vrstve) a fyzickými MAC adresami (ktoré sa používajú na linkovej vrstve). Keď zariadenie potrebuje komunikovať s iným zariadením v sieti, ale pozná len jeho IP adresu, ARP protokol umožňuje získať príslušnú MAC adresu pre správne adresovanie rámca, ktorý obsahuje dáta, ktoré majú byť prenesené.

V praxi ARP funguje tak, že zariadenie vysieľa ARP požiadavku (ARP request) s cieľovou IP adresou do siete a čaká na odpoveď. Zariadenie s cieľovou IP adresou odpovie ARP správou (ARP reply), ktorá obsahuje svoju MAC adresu. Týmto spôsobom si zariadenia v sieti vytvárajú a udržiavajú ARP tabuľku, ktorá obsahuje mapovanie medzi IP adresami a fyzickými MAC adresami.

b1 Definujte normu 568B a 568A

Normy 568A a 568B sú normy definujúce usporiadanie konektorov a farieb vodičov pre káblové pripojenie v káblových sieťach typu Ethernet. Tieto normy sa používajú pri inštalácii káblových sietí a definujú, ako sa majú káble konfigurovať, aby bolo možné dosiahnuť správnu kompatibilitu a spoľahlivé pripojenie. Hlavný rozdiel medzi nimi spočíva v usporiadaní farieb na vodičoch.

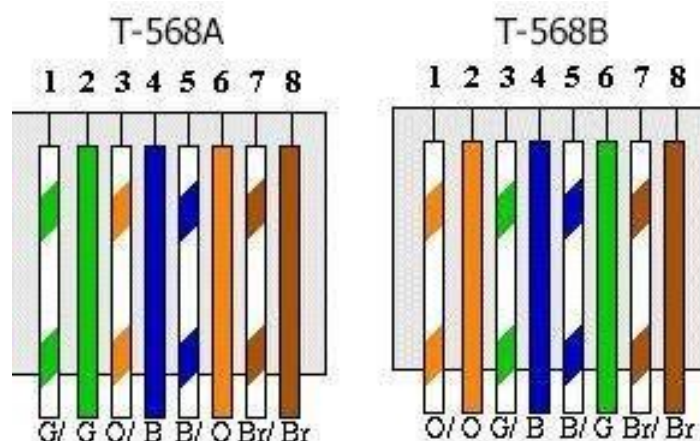
568A: Vodiče sú usporiadané nasledovne:

Biela-označená sivá / zelená	Sivá / zelená	Biela-označená oranžová	Modrá
Biela-označená modrá	Oranžová	Biela-označená hnedá / hnedá	Hnedá

568B: Vodiče sú usporiadané nasledovne:

Biela-označená oranžová	Oranžová	Biela-označená zelená	Modrá
Biela-označená modrá	Zelená	Biela-označená hnedá	Hnedá

Tieto normy sú často používané pri inštalácii Ethernetových káblových sietí, ako aj v iných typoch telekomunikačných káblových sietí. Použitie správnej normy je dôležité na dosiahnutie konzistentnosti v káblových sieťach.



b2 Popíšte rámec Ethernet

Ethernetový rámec je formát dátového rámca, ktorý sa používa na prenos údajov v sieťach Ethernet. Tento rámec je základnou jednotkou prenosu dát v rámci Ethernetu a zahŕňa rôzne polia a hlavičky, ktoré umožňujú spoľahlivú a efektívnu komunikáciu medzi zariadeniami v sieti. Tu je popis štruktúry Ethernetového rámca:

Preambula: Preambula je séria bitov, ktorá sa používa na synchronizáciu hodín prijímača s hodinami vysielacza. Je to sada opakujúcich sa bitov, ktorá pomáha prístrojom v sieti synchronizovať svoje hodiny pred príchodom samotných dát.

Start rámca (Start of Frame - SFD): Start rámca je jeden bajt, ktorý signalizuje začiatok samotných dát v rámci. SFD obsahuje určité vzory bitov, ktoré indikujú koniec preambuly a začiatok užitočných dát.

Hlavička Ethernetu: Hlavička Ethernetu obsahuje niekoľko dôležitých polí:

Cieľová MAC adresa (Destination MAC Address): Určuje MAC adresu cieľového zariadenia, pre ktoré je rámec určený.

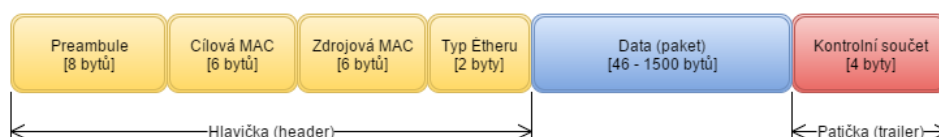
Zdrojová MAC adresa (Source MAC Address): Identifikuje MAC adresu odosielateľa rámca.

Typ alebo dĺžka (Type or Length): Toto pole označuje typ dát, ktoré nasledujú v rámci, alebo ak je menšie ako 1500, potom obsahuje dĺžku dát v rámci.

Dáta: Tieto sú samotné užitočné dáta, ktoré sa prenášajú cez sieť. Ich formát a obsah závisia od aplikačnej vrstvy a typu prenášaných údajov.

Rámcová kontrola (Frame Check Sequence - FCS): FCS je kontrolná suma, ktorá sa používa na detekciu chýb v rámci. Používa sa na kontrolu integrity dát a zabraňuje prenášaniam chybných dát cez sieť.

Ethernetový rámec sa môže líšiť v závislosti od špecifikácií Ethernetu, ako je napríklad rýchlosť prenosu dát (10 Mbps, 100 Mbps, 1 Gbps, atď.) a typ káblového média (koaxiálny kábel, UTP kábel, optické vlákno).



b3 Vysvetlite, ako switch naplňa tabuľku MAC adries, použite obrazovú prílohu č. 10

Switch naplňa svoju tabuľku MAC adries pomocou procesu nazývaného "učenie sa" (learning). Tento proces sa odohráva automaticky a umožňuje switchu vedieť, na ktorý port poslať rámce s určitými cieľovými MAC adresami.

Keď switch prijme rámec, vykoná nasledujúce kroky na naplnenie svojej MAC tabuľky:

Analýza cieľovej MAC adresy: Switch analyzuje cieľovú MAC adresu v prijatom rámci. Cieľová MAC adresa je adresa zariadenia, na ktoré je rámec určený.

Overenie MAC tabuľky: Switch sa pozrie do svojej MAC tabuľky, aby zistil, či už pozná cieľovú MAC adresu. Ak áno, zistí, na ktorom porte sa nachádza zariadenie s touto adresou.

Aktualizácia MAC tabuľky: Keď switch prijme broadcastový rámec, t.j. rámec určený pre všetky zariadenia v sieti, automaticky ho prepošle na všetky svoje porty s výnimkou portu, cez ktorý prijal rámec. Tento proces sa nazýva flooding. V tomto prípade switch zaznamená MAC adresy zdrojových zariadení týchto broadcastových rámcov a priradí ich k príslušným portom.

Ak switch nepozná cieľovú MAC adresu, čo znamená, že sa ešte nenaučil, nemôže priamo poslať rámec na príslušné zariadenie. V tomto prípade sa rámec prepošle na všetky porty okrem toho, cez ktorý bol prijatý. Tento proces sa nazýva broadcast storm. Keď zariadenie, ktoré má cieľovú MAC adresu, odpovie na tento rámec, switch sa naučí jeho MAC adresu a priradí ju k príslušnému portu v tabuľke MAC adries pre budúce komunikácie. Takto switch dynamicky aktualizuje svoju tabuľku MAC adries.

Vyčistenie zastaraných záznamov: Switch tiež pravidelne kontroluje svoju MAC tabuľku a odstraňuje z nej záznamy, ktoré sa neaktualizujú dlhšiu dobu. To zabraňuje nadmernému zapíňaniu tabuľky a udržiava ju aktuálnu.

Tento proces "učenia sa" umožňuje switchu dynamicky sa prispôbovať topológii siete a efektívne smerovať rámce na správne porty na základe ich cieľových MAC adries. To zabezpečuje efektívnu komunikáciu v sieti a minimalizuje množstvo zbytočného prenosu dát.

c1 Zaraďte počítač do majetku podniku a popíšte formy jeho opotrebenia a odpisovania v podniku

Zahrnutie počítača do majetku podniku závisí od jeho účelu a využitia. Väčšina počítačov v podnikovom prostredí sa považuje za hmotný majetok a sú zahrnuté do účtovníctva podniku. Počítač sa teda klasifikuje ako dlhodobý hmotný majetok a podlieha procesu účtovania, opotrebenia a odpisovania. Tu je súhrnný popis procesu:

Zahrnutie do majetku:

Počítač sa zahrnie do majetku podniku v čase jeho nákupu alebo akvizície. V tejto fáze sa do účtovných záznamov zaznamená jeho nákupná cena, spolu s ďalšími nákladmi spojenými s jeho nákupom, ako sú inštalačné náklady alebo poplatky za dodanie.

Počítač sa priradí konkrétnemu účtu v účtovníctve podniku, ktorý sleduje dlhodobý hmotný majetok.

Opotrebenie:

Počítač sa po čase používania opotrebováva. Opotrebenie zahŕňa fyzický úbytok hodnoty majetku v dôsledku jeho používania alebo starnutia.

Pre počítač sa vypočíta ročné opotrebenie na základe jeho očakávanej životnosti a hodnoty. Tento výpočet sa môže vykonať rôznymi metódami, ako je lineárne opotrebenie alebo metóda zrýchleného opotrebenia.

Odpisovanie:

Odpisovanie je proces, ktorým sa ročné náklady na opotrebenie rozdelia rovnomerne do účtovných období. To sa robí na účely záznamu účtovníctva a daňových odpočtov.

Odpisovanie počítača sa zaznamenáva v účtovníctve ako náklad a znižuje hodnotu majetku.

Vypočítaná hodnota odpisu sa každoročne zaznamenáva v účtovníctve podniku ako náklad a znižuje hodnotu počítača v účtovných záznamoch.

c2 Vymenujte relačné a logické operátory a vysvetlite ich význam a spôsob použitia

Relačné a logické operátory sú základné stavebné bloky pri tvorbe podmienok a výrazov v rôznych programovacích jazykoch.

Relačné operátory:

== (rovná sa): Porovnáva, či sú dve hodnoty rovnaké. Vracia hodnotu true, ak sú rovnaké, a false inak.

Príklad: `a == b` porovnáva, či hodnota `a` je rovnaká ako hodnota `b`.

!= (nerovná sa): Porovnáva, či sú dve hodnoty nerovnaké. Vracia hodnotu true, ak nie sú rovnaké, a false inak.

Príklad: `a != b` porovnáva, či hodnota `a` nie je rovnaká ako hodnota `b`.

< (menšie ako): Porovnáva, či prvá hodnota je menšia ako druhá hodnota. Vracia hodnotu true, ak je prvá hodnota menšia, a false inak.

Príklad: `a < b` porovnáva, či hodnota `a` je menšia ako hodnota `b`.

> (väčšie ako): Porovnáva, či prvá hodnota je väčšia ako druhá hodnota. Vracia hodnotu true, ak je prvá hodnota väčšia, a false inak.

Príklad: `a > b` porovnáva, či hodnota `a` je väčšia ako hodnota `b`.

<= (menšie alebo rovné): Porovnáva, či prvá hodnota je menšia alebo rovnaká ako druhá hodnota. Vracia hodnotu true, ak je prvá hodnota menšia alebo rovnaká, a false inak.

Príklad: `a <= b` porovnáva, či hodnota `a` je menšia alebo rovnaká ako hodnota `b`.

>= (väčšie alebo rovné): Porovnáva, či prvá hodnota je väčšia alebo rovnaká ako druhá hodnota. Vracia hodnotu true, ak je prvá hodnota väčšia alebo rovnaká, a false inak.

Príklad: `a >= b` porovnáva, či hodnota `a` je väčšia alebo rovnaká ako hodnota `b`.

Logické operátory:

&& (AND): Logický AND, ktorý vracia true, ak sú obe podmienky pravdivé, a false inak.

Príklad: $(a > 0) \&\& (b < 10)$ vracia true, ak je a väčšie ako 0 a zároveň b je menšie ako 10.

|| (OR): Logický OR, ktorý vracia true, ak je aspoň jedna z podmienok pravdivá, a false inak.

Príklad: $(a == 0) || (b == 0)$ vracia true, ak je buď a rovné 0, alebo b je rovné 0.

! (NOT): Logický NOT, ktorý mení hodnotu na opak. Ak bola podmienka true, stane sa false, a naopak.

Príklad: $!(a == b)$ vracia true, ak a nie je rovné b.

11. Sieťové – Smerovanie a smerovacie protokoly - Matúš

a)

1. Porovnajte statické a dynamické smerovanie (routing)
2. Zhodnoťte úlohu routera v počítačovej sieti
3. Popíšte vlastnosti PDU na sieťovej vrstve
4. Načrtnite spôsoby nastavenia komunikácie v sieti LAN
5. Porovnajte routing v IPv4 a IPv6

b)

1. Vysvetlite komunikáciu v sieti, použite obrazovú prílohu č. 11, obr. 1
2. Zdôvodnite potrebu default gateway
3. Porovnajte routovacie tabuľky, použite obrazovú prílohu č. 11, obr. 2 a 3

c)

1. Charakterizujte trh, trhové subjekty a mechanizmus, ktorý vzniká na trhu
2. Vymenujte riadiace príkazy programovacieho jazyka, vysvetlite a predvedte ich použitie

a1 Porovnajte statické a dynamické smerovanie (routing)

Statické a dynamické smerovanie sú dva základné prístupy k rozhodovaniu o tom, ako smerovať pakety v sieťach. Tu je porovnanie oboch prístupov:

Statické smerovanie (manuálne nastavené):

Statické smerovanie sa nastavuje manuálne administrátorom siete. Každý router alebo prepínač má definované statické smerovacie tabuľky, ktoré určujú, ako majú byť pakety smerované.

Stabilita:

Statické smerovanie je stabilné a predvídateľné, pretože smerovacie tabuľky sa nemenné, pokiaľ ich administrátor ručne nezmení.

Jednoduchosť:

Implementácia statického smerovania je jednoduchá a nenáročná, pretože nevyžaduje žiadne špeciálne protokoly alebo dynamické aktualizácie.

Vhodné pre malé siete:

Statické smerovanie sa často používa v malých sieťach alebo v jednoduchých topológiách, kde je malý počet routerov a stabilná sieťová topológia.

Dynamické smerovanie (automaticky aktualizované):

Dynamické smerovacie protokoly ako napríklad RIP, OSPF alebo EIGRP automaticky výmenou informácií medzi susednými routermi. Tieto informácie sa používajú na automatické aktualizácie smerovacích tabuliek.

Adaptabilita:

Dynamické smerovacie protokoly sú schopné dynamicky reagovať na zmeny v sieti, ako sú výpadky linky alebo zmeny topológie. Tým sa zabezpečuje, že smerovacie tabuľky sú vždy aktuálne.

Komplexnosť:

Implementácia dynamického smerovania je komplexnejšia ako statické smerovanie, pretože vyžaduje konfiguráciu a správu smerovacích protokolov.

Vhodné pre veľké siete:

Dynamické smerovanie sa najčastejšie používa v rozsiahlych sieťach, kde je veľký počet routerov a rýchlo sa meniacia topológia.

V závislosti od veľkosti siete, stability a požiadaviek na dynamiku sa vyberie jeden z týchto prístupov alebo ich kombinácia. Vo väčších sieťach sa často používa dynamické smerovanie, zatiaľ čo v menších sieťach môže byť statické smerovanie vhodnejšie.

a2 Zhodnotte úlohu routra v počítačovej sieti

Router je základným prvkom v počítačovej sieti a hrá kľúčovú úlohu pri prenose dát medzi rôznymi sieťovými segmentmi. Tu je niekoľko hlavných funkcií a úloh, ktoré router zabezpečuje v počítačovej sieti:

Smerovanie paketov:

Hlavnou úlohou routera je smerovanie paketov medzi rôznymi sieťovými segmentmi alebo medzi rôznymi sieťami. To zahŕňa rozhodovanie o optimálnych cestách na základe dostupných smerovacích informácií a topológie siete.

Rozhrania sietí:

Router má niekoľko rozhraní sietí, ktoré umožňujú pripojenie k rôznym sieťovým segmentom. Tieto rozhrania môžu byť fyzické alebo virtuálne a môžu podporovať rôzne typy sieťových protokolov.

Firewall a bezpečnosť:

Mnohé routery poskytujú funkcie firewallu a zabezpečenia siete, ktoré umožňujú filtrovanie a monitorovanie sieťového toku. Tieto funkcie môžu chrániť sieť pred neoprávneným prístupom a útokmi.

NAT (Network Address Translation):

Router môže vykonávať funkciu NAT, ktorá umožňuje prevádzku súkromných IP adries v sieti a ich mapovanie na verejné IP adresy, ktoré sú použiteľné na internete.

QoS (Quality of Service):

Niektoré routery poskytujú funkcie QoS, ktoré umožňujú riadenie šírky pásma a prioritizáciu dátového toku. To môže byť užitočné pre správu siete a optimalizáciu výkonu.

Distribúcia dátových správ:

Router môže tiež distribuovať dátové správy medzi rôznymi uzlami v sieti na základe smerovacích informácií a pravidiel smerovania.

Celkovo povedané, router je kľúčovým prvkom v počítačovej sieti, ktorý umožňuje efektívnu komunikáciu medzi rôznymi uzlami v sieti a poskytuje zabezpečenie, správu siete a optimalizáciu prenosu dát.

a3 Popíšte vlastnosti PDU na sieťovej vrstve

PDU (Protocol Data Unit) na sieťovej vrstve je dátová štruktúra, ktorá sa používa na prenos dát cez sieť na základe IP protokolu. Na sieťovej vrstve sa PDU nazýva packet.

IP Hlavička: Každá PDU na sieťovej vrstve obsahuje IP hlavičku, ktorá obsahuje rôzne polia, ako sú zdrojová a cieľová IP adresa, verzia IP, typ služby, dĺžka dát, identifikátor fragmentu a kontrolná suma.

Adresácia: PDU na sieťovej vrstve používa logickú adresáciu na identifikáciu zdroja a cieľa dát. Táto adresa je známa ako IP adresa a je jedinečná pre každé zariadenie v sieti.

Fragmentácia: Ak sú dáta väčšie ako maximálna dátová jednotka (MTU) na linkovej vrstve, môžu byť rozdelené na menšie fragmenty, ktoré sa prenášajú samostatne a neskôr znovu zostavené na cieľovom zariadení.

Smerovanie: PDU na sieťovej vrstve je smerovaná cez sieť na základe IP adries v jej hlavičke. Každý router v sieti používa tieto adresy na rozhodovanie o tom, ako preposlať PDU na ďalší hop smerom k cieľovej adrese.

Spoľahlivosť: Sieťová vrstva neposkytuje žiadnu garantovanú spoľahlivosť doručenia dát. To znamená, že PDU môže byť stratená, opakovaná alebo doručená v nesprávnom poradí. Spoľahlivosť je zvyčajne zabezpečená nad sieťovou vrstvou pomocou protokolov transportnej vrstvy ako TCP.

Nezávislosť na médiu: PDU na sieťovej vrstve je nezávislá na fyzickom médiu, cez ktoré je prenášaná. Môže byť prenášaná cez rôzne typy médií, ako sú Ethernet, WiFi, optické vlákno a iné.

a4 Načrtnite spôsoby nastavenia komunikácie v sieti LAN

V sieti **LAN (Local Area Network)** existuje niekoľko spôsobov nastavenia komunikácie medzi zariadeniami.

Statické nastavenie IP adries:

Pri statickom nastavení každé zariadenie (počítač, tlačiareň, server atď.) má pridelenú statickú IP adresu. Tieto IP adresy sú ručne konfigurované na každom zariadení a musia byť jedinečné v rámci siete. Tento prístup je vhodný pre menšie siete s malým počtom zariadení, ktoré sa zriedka menia.

Dynamic Host Configuration Protocol (DHCP):

DHCP je protokol, ktorý umožňuje zariadeniam dynamicky získať IP adresu a ďalšie konfiguračné informácie (ako sú maska siete, brána a DNS servery) od DHCP servera v sieti. Tento prístup je vhodný pre väčšie siete, kde je potrebné efektívne riadiť pridelenie IP adries.

Automatické konfigurácie linkovej vrstvy:

Niektoré siete LAN môžu byť nakonfigurované tak, aby používali automatické konfigurácie linkovej vrstvy, ako je napríklad protokol Link Layer Discovery Protocol (LLDP). Tento protokol umožňuje zariadeniam získať informácie o susedných zariadeniach v sieti, ako sú ich identifikátory a funkcie.

Nastavenie prístupových práv:

Nastavenie prístupových práv definuje, ktoré zariadenia majú prístup k rôznym zdrojom a službám v sieti LAN. Tieto pravidlá môžu byť konfigurované pomocou firewallu, sieťových politík alebo konkrétnych nastavení zariadenia (napríklad prístupových kontrolných zoznamov).

Konfigurácia sieťových služieb:

Konfigurácia sieťových služieb ako DNS (Domain Name System), DHCP (Dynamic Host Configuration Protocol), smerovania a iných služieb je dôležitou súčasťou nastavenia komunikácie v sieti LAN. Tieto služby umožňujú zariadeniam efektívne komunikovať a prístupovať k zdieľaným zdrojom a službám v sieti.

a5 Porovnajte routing v IPv4 a IPv6

Routing v IPv4 a IPv6 sa líši v niekoľkých kľúčových aspektoch. Tu sú niektoré hlavné rozdiely:

Veľkosť adresného priestoru:

Jedným z najvýraznejších rozdielov medzi IPv4 a IPv6 je veľkosť adresného priestoru. IPv4 používa 32-bitové adresy, čo poskytuje približne 4,3 miliardy možných adries. Na druhej strane IPv6 používa 128-bitové adresy, čo poskytuje astronomicky veľký priestor s približne $3,4 \times 10^{38}$ (340 undecillion) možných adries. 2 na 128 IPv6 existuje.

Formát adries:

Formát adries IPv4 je zložený z desiatich základných čísel od 0 do 255 (napr. 192.168.1.1), zatiaľ čo formát adries IPv6 je hexadecimálny a pozostáva z 8 skupín štyroch hexadecimálnych číslic oddelených dvojbodkami (napr. 2001:0db8:85a3:0000:0000:8a2e:0370:7334).

Podpora pre auto-konfiguráciu:

IPv6 má zabudovanú podporu pre auto-konfiguráciu založenú na protokole Neighbor Discovery Protocol (NDP). Táto funkcia umožňuje zariadeniam získať svoju IPv6 adresu automaticky bez potreby DHCP servera.

Riešenie problému adresného vyčerpania:

Adresný priestor IPv4 je takmer vyčerpaný kvôli obrovskému nárastu počtu pripojených zariadení. IPv6 má oveľa väčší adresný priestor, čo pomáha riešiť tento problém vyčerpania adries.

Jednoduchosť smerovania:

IPv6 má zjednodušený hlavičkový formát a obsahuje len základné informácie potrebné pre smerovanie. To zjednodušuje proces smerovania v porovnaní s IPv4.

Podpora pre QoS (Quality of Service):

IPv6 obsahuje zabudovanú podporu pre kvalitu služby (QoS), ktorá umožňuje prioritizáciu dátového toku na základe ich dôležitosti. Táto funkcia je užitočná pre aplikácie s vysokými nárokmi na spoľahlivosť a nízku oneskorenie.

V súhrne, IPv6 prináša mnoho výhod oproti IPv4 vrátane väčšieho adresného priestoru, lepšej podpory pre auto-konfiguráciu, zjednodušeného smerovania a podpory pre QoS. Prechod na IPv6 postupne prebieha ako riešenie problémov spojených s vyčerpaním adresného priestoru a potrebou pre lepšiu podporu moderných aplikácií a služieb v sieťach.

b1 Vysvetlite komunikáciu v sieti, použite obrazovú prílohu č. 11, obr. 1

Komunikácia v sieti je proces výmeny dát medzi dvoma alebo viacerými zariadeniami v sieti. Tento proces môže byť realizovaný rôznymi spôsobmi a môže zahŕňať rôzne protokoly a technológie. Tu je všeobecný pohľad na komunikáciu v sieti:

Identifikácia a adresovanie:

Každé zariadenie v sieti má svoju jedinečnú identifikáciu, ktorá mu umožňuje komunikovať s inými zariadeniami. Táto identifikácia sa väčšinou vykonáva pomocou IP adres na sieťovej vrstve a MAC adres na linkovej vrstve.

Naviazanie spojenia:

Pred samotnou výmenou dát musia zariadenia naviazať spojenie. Tento proces môže zahŕňať naviazanie spojenia na fyzickej vrstve pomocou technológií ako je Ethernet alebo Wi-Fi, ako aj naviazanie spojenia na vyšších vrstvách pomocou protokolov ako TCP (Transmission Control Protocol) alebo UDP (User Datagram Protocol).

Výmena dát:

Po úspešnom naviazaní spojenia môžu zariadenia začať výmenu dát. Tento proces môže zahŕňať prenos rôznych typov dát, vrátane textových, audiovizuálnych, obrázkových a iných dát.

Spracovanie a správa dát:

Po prijatí dát zariadenie ich spracuje podľa potreby. To môže zahŕňať dekodovanie, analýzu, ukladanie do pamäte alebo ďalšie spracovanie na základe určitých pravidiel alebo algoritmov.

Odpoveď a potvrdenie:

Po spracovaní dát môže zariadenie zaslať odpoveď alebo potvrdenie o prijatí dát späť k zdrojovému zariadeniu. Toto potvrdenie môže byť dôležité pre zabezpečenie spoľahlivosti komunikácie.

Uzavretie spojenia:

Keď je výmena dát dokončená, môžu zariadenia uzavrieť spojenie. To môže byť realizované prostredníctvom formálneho uzavretia spojenia na vyšších vrstvách protokolu alebo prostredníctvom fyzického odpojenia na nižších vrstvách.

b2 Zdôvodnite potrebu default gateway

Default gateway (predvolená brána) je dôležitý prvok v počítačových sieťach, a to z niekoľkých dôvodov:

Smerník pre komunikáciu mimo lokálnej siete: Default gateway umožňuje zariadeniam v lokálnej sieti komunikovať s inými sieťami mimo svojej lokálnej siete. Keď zariadenie potrebuje komunikovať s cieľovým zariadením nachádzajúcim sa mimo jeho lokálnej siete, posunie paket na default gateway, ktorý sa potom postará o smerovanie paketov do správnej cieľovej siete.

Prístup na internet: Väčšina sietí používa default gateway na prístup na internet. Keď sa počítač pokúša pripojiť k internetu, posielané dátové pakety musia byť adresované na internetovú bránu, ktorá smeruje dáta do verejnej siete.

Nastavenie bezpečnosti a smerovania: Default gateway môže byť použitý aj na nastavenie bezpečnostných politík a filtrovania dát. To umožňuje správcovi siete riadiť, ktoré typy dát môžu a nemôžu opustiť sieť.

Umožňuje zdieľanie zdrojov: Default gateway môže slúžiť aj ako miesto, kde sú zdieľané zdroje ako tlačiarne, úložisko alebo server umiestnené mimo lokálnej siete, a umožňuje zariadeniam v tejto sieti prístup k týmto zdieľaným zdrojom.

Celkovo povedané, default gateway je nevyhnutnou súčasťou siete, ktorá umožňuje zariadeniam v lokálnej sieti komunikovať s ostatnými sieťami a poskytuje prístup na internet, zdieľanie zdrojov a implementáciu bezpečnostných politík.

[b3 Porovnajzte routovacie tabuľky, použite obrazovú prílohu č. 11, obr. 2 a 3](#)

Routovacie tabuľky (routing tables) sú kľúčovými komponentami smerovania v počítačových sieťach. Porovnajme ich:

IPv4 vs IPv6:

Routovacie tabuľky pre IPv4 obsahujú záznamy pre IPv4 siete a sú zvyčajne štruktúrované pomocou adries siete a masky siete. Na druhej strane, routovacie tabuľky pre IPv6 obsahujú záznamy pre IPv6 siete, ktoré sú reprezentované pomocou IPv6 adries.

Veľkosť:

Routovacie tabuľky IPv4 sú obvykle menšie ako tabuľky IPv6, pretože IPv4 má obmedzený adresný priestor. IPv6 má oveľa väčší adresný priestor, takže sú potrebné väčšie tabuľky na udržanie informácií o dostupných sieťach.

Typy záznamov:

Routovacie tabuľky môžu obsahovať rôzne typy záznamov, vrátane záznamov o smerovanie, ktoré definujú, ako smerovať pakety na cieľové siete, a záznamov o rozhraniach, ktoré definujú, cez ktoré rozhranie sa majú pakety poslať na cieľ.

Dynamické vs statické záznamy:

Routovacie tabuľky môžu obsahovať dynamické záznamy, ktoré sú získavané pomocou dynamických smerovacích protokolov ako napríklad OSPF alebo BGP, alebo statické záznamy, ktoré sú ručne konfigurované administrátorom siete.

OSPF (Open Shortest Path First):

OSPF je smerovací protokol typu link-state, ktorý sa používa pre smerovanie dát v intranetových a interných sieťach.

Každý router v sieti vytvára mapu topológie siete, kde sú uložené informácie o všetkých routeroch a spojeniach v sieti.

Na základe týchto informácií OSPF vypočíta najkratšiu cestu (shortest path) k cieľovej sieti pomocou algoritmu Dijkstra.

OSPF je vhodný pre stredné a veľké siete a je schopný dynamicky reagovať na zmeny topológie siete.

Priorita záznamov:

V routovacích tabuľkách môžu byť záznamy usporiadané podľa priority. Záznamy s nižšou metrikou alebo lepším rozsahom sú preferované pred záznamami s vyššou metrikou alebo horším rozsahom.

Správa a údržba:

Routovacie tabuľky vyžadujú pravidelnú údržbu a správu, aby boli aktualizované a efektívne. To zahŕňa pridávanie, aktualizovanie a odstránenie záznamov podľa potreby, aby sa zabezpečilo optimálne smerovanie paketov v sieti

Zhrnutie: Rozdiel medzi "directly connected" a "received" v kontexte IPv6 je, že "directly connected" označuje rozhranie, ktoré je priamo pripojené k určitej sieti, zatiaľ čo "received" označuje stav prijatia dátového paketu na danom rozhraní. Oba pojmy sa používajú na opisovanie správania sieťových rozhraní

c1 Charakterizujte trh, trhové subjekty a mechanizmus, ktorý vzniká na trhu

Trh je miesto, kde dochádza k stretu dopytu a ponuky tovarov a služieb medzi rôznymi subjektmi v ekonomike.

Subjekty: firmy, domácnosti, štát

Funkcie:

firmy – predávajúci (tovary, služby), kupujúci (výrobné faktory)

domácnosti – predávajúci (výrobné faktory), kupujúci (tovary a služby)

štát – špecifický subjekt, vstupuje na trh vždy, ak je potrebné odstrániť negatívne javy, monopoly, chrániť obyvateľstvo = minimálna mzda.

Typy trhov:

Čiastkový – len druh tovaru

Agregátny – všetky tovary a služby

Trh výrobných faktorov – predaj a kúpa VF – pôda, práca, kapitál

Finančný trh – peňažný, banky a burzy

Národný – trh s automobilmi SR

Medzinárodný – EU

Svetový – medzinárodný obchod

Trhový mechanizmus

Dopyt a ponuka: Trh je charakterizovaný existenciou dopytu a ponuky. Dopyt predstavuje množstvo tovarov alebo služieb, ktoré spotrebitelia sú ochotní kúpiť za určitú cenu, zatiaľ čo ponuka predstavuje množstvo tovarov alebo služieb, ktoré sú k dispozícii na predaj zo strany výrobcov alebo poskytovateľov služieb.

Ceny: Na trhu sa stanovujú ceny na základe interakcie medzi dopytom a ponukou. Vysoký dopyt v kombinácii s nízkou ponukou môže viesť k zvýšeniu cien a naopak. Ceny na trhu slúžia ako mechanizmus, ktorý reguluje dopyt a ponuku a pomáha udržiavať rovnováhu na trhu.

Konkurencia: Trhy sú často charakterizované existenciou konkurencie medzi rôznymi subjektmi. Konkurencia podporuje efektívnosť a inovácie tým, že podnecuje firmy k lepšiemu poskytovaniu tovarov a služieb za konkurencieschopné ceny.

Mechanizmus trhu: Mechanizmus trhu je spôsob, akým sa dopyt a ponuka stretávajú a interagujú na trhu. Tento mechanizmus zahŕňa proces stanovenia cien, vyjednávanie, uzatváranie obchodov a prerozdelenie zdrojov v ekonomike podľa potrieb a preferencií spotrebiteľov.

c2 Vymenujte riadiace príkazy programovacieho jazyka, vysvetlite a predvedte ich použitie

If-else príkaz:

Príkaz if-else umožňuje programu vykonávať rôzne časti kódu v závislosti na podmienke.

```
int x = 10;
if (x > 5)
{
    Console.WriteLine("x je väčšie ako 5");
}
else
{
    Console.WriteLine("x je menšie alebo rovné 5");
}
```

Cyklus for:

Cyklus for sa používa na iteráciu cez kolekciu prvkov (ako je pole) a vykonáva určité operácie pre každý prvok.

```
int[] numbers = { 1, 2, 3, 4, 5 };
for (int i = 0; i < numbers.Length; i++)
{
    Console.WriteLine(numbers[i]);
}
```

Cyklus while:

Cyklus while opakuje vykonávanie bloku kódu, kým je splnená určitá podmienka.

```
int i = 0;
while (i < 5)
{
    Console.WriteLine(i);
    i++;
}
```

Príkaz break:

Príkaz break sa používa na prerušenie cyklu, keď je splnená určitá podmienka.

```
int[] numbers = { 1, 2, 3, 4, 5 };
foreach (int num in numbers)
{
    if (num == 3)
    {
        break;
    }
    Console.WriteLine(num);
}
```

12. Sieťové – Diagnostika sietí - Paťo

a)

1. Vysvetlite zásady diagnostiky problémov v počítačových sieťach na RM OSI a TCP/IP
2. Vymenujte základné softvérové nástroje pre diagnostiku počítačovej siete
3. Načrtnite spôsoby nastavenia požadovaných parametrov na routeri (meno, IP adresy, konzola, ...)
4. Popíšte postup pre overenie konektivity ľubovoľného koncového zariadenia v počítačovej sieti
5. Uveďte postup pre zobrazenie nastavených parametrov na sieťovom zariadení

b)

1. Vysvetlite dôvody a spôsoby kryptovania hesiel na sieťových zariadeniach
2. Popíšte možnosti diaľkového prístupu k sieťovým zariadeniam
3. Vysvetlite význam protokolu ARP

c)

1. Popíšte diagnostiku a odstraňovanie porúch v navrhutej počítačovej sieti, použite obrazovú prílohu č. 12
2. Charakterizujte poľa, predvedte deklaráciu a vytvorenie poľa aj z hľadiska pamäte

a1 Vysvetlite zásady diagnostiky problémov v počítačových sieťach na RM OSI a TCP/IP

○ Zásady diagnostiky problémov v sieťach na základe modelu ISO/OSI:

1. **Rozdelenie funkcií:** Model OSI rozdeľuje funkcie siete do siedmich vrstiev. Pri diagnostike problému sa každá vrstva skúma individuálne, aby sa zistilo, v ktorej vrstve nastal problém.
2. **Zásada oddelenia:** Každá vrstva má jasne definované rozhranie a zodpovedá len za svoju vlastnú funkciu. Pri diagnostike je dôležité oddeliť problémy na úrovni jednotlivých vrstiev a minimalizovať prekrývajúce sa problémy.
3. **Postupné testovanie:** Diagnostika začína na najvyššej vrstve a postupuje sa smerom nadol. To znamená, že najskôr sa skúma aplikácia a postupne sa presúva nižšie k fyzickej vrstve.
4. **Protokolová analýza:** Pri diagnostike sa často používa analýza sieťových protokolov na jednotlivých vrstvách. Toto umožňuje identifikovať chyby v komunikácii medzi jednotlivými zariadeniami.

○ Zásady diagnostiky problémov v sieťach na základe modelu TCP/IP:

1. **Zjednodušená štruktúra:** Model TCP/IP kombinuje niektoré funkcie, ktoré sú oddelené v modeli OSI. Napríklad, aplikačná vrstva v TCP/IP modeli obsahuje niektoré funkcie, ktoré sú rozdelené medzi aplikačnú, prezentačnú a relačnú vrstvu v modeli OSI.
2. **Zameranie na komunikáciu:** Diagnostika problémov v sieťach na základe modelu TCP/IP sa často zameriava na overenie konektivity medzi zariadeniami a správnosť konfigurácie siete.
3. **Monitoring sieťových prúdov:** Pri diagnostike v TCP/IP sieti sa často používa monitoring sieťových prúdov (napríklad pomocou nástrojov ako Wireshark) na analýzu komunikácie medzi zariadeniami a identifikáciu možných problémov.
4. **Rozšírená znalosť protokolov:** Vzhľadom na to, že model TCP/IP je výrazne používaný v súčasných sieťach, diagnostika problémov vyžaduje rozsiahle znalosti o rôznych protokoloch TCP/IP, ako aj ich správaní a možných chybových stavoch.

a2 Vymenujte základné softvérové nástroje pre diagnostiku počítačovej siete

- ping – príkaz na atestovanie dostupnosti pingnutého zariadenia
- tracert – sleduje trasu ktorou prechádza (IP adresy)
- telnet – používa sa na ovládanie počítača na diaľku
- netstat – nástroj na zobrazenie aktívnych spojení a štatistík prechodu dát, Ukazuje aktívne spojenia, rozhrania siete, tabuľky rozhraní a iné štatistiky

a3 Načrtnite spôsoby nastavenia požadovaných parametrov na routeri (meno, IP adresy, konzola, ...)

1. Nastavenie názvu routeru (hostname):

- Prístup k príkazovému riadku routeru cez sériový kábel alebo sieťové pripojenie (SSH/Telnet).
- Prepnutie do režimu konfigurácie pomocou príkazu enable.
- Vstup do režimu konfigurácie globálneho nastavenia príkazom configure terminal.

- Nastavenie názvu routeru pomocou príkazu hostname <názov>.
- Uloženie zmien príkazom copy running-config startup-config.

2. Nastavenie IP adresy rozhrania (interface):

- Prepnutie do režimu konfigurácie pomocou príkazu configure terminal.
- Vstup do režimu konfigurácie rozhrania príkazom interface <rozhranie>.
- Nastavenie IP adresy na rozhraní príkazmi ip address <adresa> <maska> pre IPv4 alebo ipv6 address <adresa>/<dlzka_prefixu> pre IPv6.
- Aktivácia rozhrania príkazom no shutdown.
- Uloženie zmien príkazom copy running-config startup-config.

3. Nastavenie prístupu cez konzolu:

- Prepnutie do režimu konfigurácie pomocou príkazu configure terminal.
- Nastavenie hesla pre prístup k konzole príkazom line console 0 a príkazom password <heslo>.
- Aktivácia autentifikácie príkazom login.
- Uloženie zmien príkazom copy running-config startup-config.

a4 Popíšte postup pre overenie konektivity ľubovoľného koncového zariadenia v počítačovej sieti

1. Zistíte IP adresu koncového zariadenia: Ak máte pripojené ľubovoľné koncové zariadenie, ako je napríklad počítač alebo iné sieťové zariadenie, zistíte jeho pridelenú IP adresu. Môžete to urobiť pomocou príkazu ipconfig (v prípade systému Windows) alebo ifconfig (v prípade systému Linux/macOS) v príkazovom riadku.

2. Pingovanie koncového zariadenia: Použite príkaz ping spolu s IP adresou koncového zariadenia na overenie jeho dostupnosti v sieti. Napríklad, ak IP adresa koncového zariadenia je 192.168.1.100, môžete použiť príkaz ping 192.168.1.100 v príkazovom riadku. Pokiaľ dostanete odpoveď od koncového zariadenia (napríklad "Reply from 192.168.1.100: bytes=32 time<1ms TTL=128"), znamená to, že je dostupné v sieti.

3. Kontrola firewallu a nastavení siete na koncovom zariadení: Ak neprijímate odpovede na pingovacie požiadavky, môže to byť spôsobené nastaveniami firewallu na koncovom zariadení alebo neplatnými nastaveniami siete (napríklad nesprávna IP adresa, maska podsiete, brána alebo DNS server). Skontrolujte tieto nastavenia a uistite sa, že sú správne nakonfigurované.

4. Skontrolujte fyzické pripojenie: Uistite sa, že koncové zariadenie je správne pripojené k sieti. Skontrolujte, či je sieťový kábel korektne zapojený do zariadenia a do sieťovej zásuvky alebo prepojenia.

5. Skontrolujte sieťové zariadenia medzi zdrojom a cieľom: Ak aj po pingovaní koncového zariadenia nezískate odpoveď, môže to naznačovať problém v sieti medzi zdrojom a cieľom. Skontrolujte, či sú sieťové zariadenia (napríklad switch, router) medzi zdrojom a cieľom správne nakonfigurované a či nie sú problémy s ich funkčnosťou.

a5 Uved'te postup pre zobrazenie nastavených parametrov na sieťovom zariadení

1. Zobrazenie všetkých konfiguračných parametrov: show running-config
2. Zobrazenie základných informácií o rozhraniach: show interfaces
3. Zobrazenie informácií o smerovaní: show ip route pre IPv4 alebo show ipv6 route pre IPv6
4. Zobrazenie tabuľky ARP: show arp

b1 Vysvetlite dôvody a spôsoby kryptovania hesiel na sieťových zariadeniach

- **Dôvody pre kryptovanie hesiel:**

1. **Ochrana pred neoprávneným prístupom:** Heslá sú základným mechanizmom autentifikácie, ktorý umožňuje určiť, kto má oprávnenie pristupovať k sieťovým zariadeniam a konfiguráciám. Kryptovanie hesiel zabraňuje neoprávneným osobám v čítaní hesiel z konfiguračných súborov alebo ich odchyťovaní počas prenosu cez sieť.
2. **Zabezpečenie citlivých údajov:** Mnohé konfiguračné informácie na sieťových zariadeniach obsahujú citlivé údaje, ako sú prihlasovacie údaje k iným zariadeniam alebo k dôležitým službám. Kryptovanie hesiel zabraňuje ich zneužitiu a chráni dôvernosť týchto informácií.
3. **Dodržiavanie bezpečnostných noriem a regulácií:** Mnohé bezpečnostné normy a regulácie vyžadujú zabezpečenie hesiel pomocou kryptografických metód na ochranu sietí a zariadení pred neoprávneným prístupom.

- **Spôsoby kryptovania hesiel:**

1. **Hashovanie hesiel:** Heslá môžu byť uložené v konfiguračných súboroch vo forme hashov. Hashovanie hesiel znamená, že heslo je transformované na nečitateľnú postupnosť znakov pomocou kryptografickej hashovej funkcie. Pri overovaní hesla sa zadané heslo tiež zahashuje a porovná sa s uloženým hashom.
2. **Kryptografické algoritmy:** Na zabezpečenie hesiel sa používajú rôzne kryptografické algoritmy, ako napríklad MD5, SHA-1, SHA-256 a iné. Tieto algoritmy slúžia na transformáciu hesiel do formy, ktorá je veľmi ťažko alebo nemožno rekonštruovať

b2 Popíšte možnosti diaľkového prístupu k sieťovým zariadeniam

1. **SSH (Secure Shell):**

- SSH je veľmi bezpečný protokol, ktorý umožňuje šifrovanú komunikáciu medzi klientom a serverom.
- Pomocou SSH je možné vytvoriť diaľkové spojenie k sieťovým zariadeniam, ako sú routery, prepínače, servery, a pracovať s nimi z príkazového riadku.
- Poskytuje autentifikáciu pomocou používateľských mien a hesiel alebo pomocou kľúčov.

2. **Telnet:**

- Telnet je protokol, ktorý umožňuje jednoduché, ale nešifrované diaľkové pripojenie k sieťovým zariadeniam.
- Poskytuje prístup k príkazovému riadku zariadenia rovnako ako SSH, ale komunikácia medzi klientom a serverom nie je šifrovaná, čo môže predstavovať bezpečnostné riziko

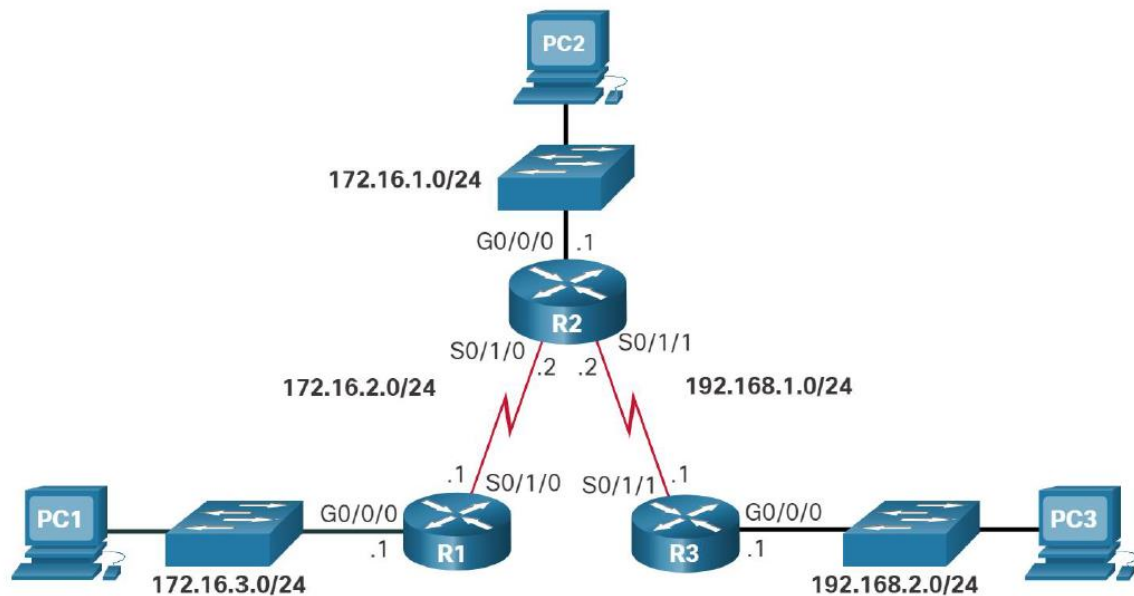
b3 Vysvetlite význam protokolu ARP

- **ARP (Address Resolution Protocol)** je protokol, ktorý slúži na mapovanie IP adries na fyzické MAC adresy v lokálnej sieti. Jeho hlavným cieľom je umožniť zariadeniam v sieti nájsť fyzickú MAC adresu zariadenia, ktorému chcú poslať pakety v rámci tej istej lokálnej siete.
- **Význam protokolu ARP** je neoddeliteľný od správnej komunikácie v lokálnej sieti. Bez správnej mapovania IP adries na fyzické MAC adresy by bolo nemožné správne doručiť pakety v rámci siete. ARP protokol je tak kľúčovým prvkom v zabezpečení efektívnej a spoľahlivej komunikácie v lokálnej sieti.

c1 Popíšte diagnostiku a odstraňovanie porúch v navrhnujej počítačovej sieti, použite obrazovú prílohu č. 12

Text NENÍ

Obrazová príloha k zadaniu č. 12



c2 Charakterizujte polia, predved'te deklaráciu a vytvorenie poľa aj z hľadiska pamäte

- Polia sú dátové štruktúry v programovaní, ktoré umožňujú ukladať viacero hodnôt rovnakého typu pod jedným menom. Každá hodnota v poli je priradená k jednému alebo viacerým indexom, ktoré sa používajú na identifikáciu konkrétnej hodnoty v poli.
- Deklarácia a vytvorenie poľa:
 - Deklarácia poľa celých čísel s veľkosťou 5:

```
int[] poleCelychCisel = new int[5];
```

Najskôr prebehne deklarácia poľa, čo je časť za priradením „=“ teda `int[] poleCelychCisel` je deklarácia poľa. `new int[5]` je vytvorenie int-ového poľa s maximálne 5 prvkami v ňom.

- Deklarácia poľa reťazcov s veľkosťou 3:

```
string[] poleRetazcov = new string[3];
```

- Deklarácia prázdneho poľa a následné naplnenie:

```
int[] poleCelychCisel = new int[5];  
for (int i = 0; i < poleCelychCisel.Length; i++)  
{  
    poleCelychCisel[i] = i + 10; // Naplnenie poľa hodnotami (0, 10, 20, 30...)  
}
```


13. Sieťové – Komunikácia na vrstve L2 - Majo

a)

1. Popíšte funkciu prepínača (switch)
2. Vysvetlite pojem metóda kontroly k prístupu na médium
3. Porovnajzte funkciu LLC a MAC podvrstvy
4. Vysvetlite význam virtuálnych lokálnych sietí (VLAN)
5. Načrtnite štruktúru rámca (frame)

b)

1. Popíšte spôsoby prístupu na médium, použite obrazovú prílohu č. 13
2. Načrtnite schému zapojenia troch switchov pripojených k smerovaču (routru)
3. Charakterizujte cyklický vývoj trhovej ekonomiky

c)

1. Zhodnoťte dôsledky cyklického vývoja ekonomiky (inflácia, nezamestnanosť)
2. Vysvetlite čo sú to metódy a funkcie, vytvorte vlastnú ľubovoľnú metódu a predved'te jej zovolanie

a1 Popíšte funkciu prepínača (switch)

Switch je sieťový prvok, ktorý má za úlohu prepájať koncové zariadenia alebo časti siete. Práve switch je zariadenie, o ktorom môžeme vyhlásiť, že vytvára sieť, pretože stojí v ich strede. Koordinuje sieťovú prevádzku tak, že prijaté dátové rámce odosiela výlučne k ich adresátovi, čím vyťažuje pripojené zariadenia individuálne.

a2 Vysvetlite pojem metóda kontroly k prístupu na médium

Metóda kontroly k prístupu na médium je spôsob, ako riadiť, kto má povolenie na prístup k určitému médiu, ako napríklad sieť, súbory alebo iné prostriedky. Tieto metódy môžu zahŕňať heslá, biometrické údaje, tokeny alebo iné formy overenia identity, aby sa zabezpečilo, že len oprávnení používatelia majú prístup.

a3 Porovnajzte funkciu LLC a MAC podvrstvy

LLC

- Riadi logickú komunikáciu medzi zariadeniami.
- Zabezpečuje spoľahlivú komunikáciu cez fyzické médiá.
- Špecifikuje protokoly na adresovanie, riadenie toku a zabezpečenie spoľahlivosti prenosu údajov.

MAC

- Riadi prístup k fyzickému médiu, ako je napríklad Ethernet.
- Rieši konflikty pri súčasnom prístupe viacerých zariadení k médiu.

-Stanovuje pravidlá pre zariadenia na získavanie prístupu k médiu a riešenie kolízií, ako je napríklad CSMA/CD pre Ethernet.

LLC a MAC sú dve podvrstvy v rámci vrstvy prenosu údajov v modeli OSI. Základný rozdiel spočíva v tom, že LLC pracuje s logickými spojeniami a riadi spoľahlivosť prenosu údajov, zatiaľ čo MAC sa zaoberá fyzickým médiom a riadi prístup k nemu. Spolu tieto podvrstvy umožňujú efektívnu a spoľahlivú komunikáciu medzi zariadeniami v sieti.

a4 Vysvetlite význam virtuálnych lokálnych sietí (VLAN)

Virtuálne lokálne siete (VLAN) sú koncept, ktorý umožňuje rozdelenie fyzickej siete na viacero logických sietí na základe potreby alebo organizačnej štruktúry.

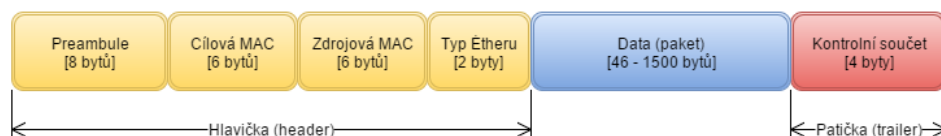
Hlavný význam VLAN spočíva v izolácii a segmentácii sieťovej prevádzky, čo umožňuje:

- 1. Bezpečnosť** - VLAN môžu izolovať dôležité údaje alebo služby od ostatnej sieťovej premávky (trafficu), čím sa znižuje riziko neoprávneného prístupu.
- 2. Optimalizácia sieťového prevádzkovania** - Použitím VLAN môžu organizácie segmentovať sieťový provoz podľa potrieb alebo oddelenia, čo umožňuje efektívnejšiu správu a optimalizáciu prevádzky.
- 3. Flexibilita** - VLAN umožňujú organizáciám rýchlo meniť alebo prispôbovať sieťovú topológiu podľa zmenených potrieb bez potreby fyzických zmien na hardvéri.
- 4. Organizačná štruktúra** - VLAN môžu odrážať organizačnú štruktúru firmy, čo umožňuje lepšiu správu a riadenie prístupu k sieťovým zdrojom.

a5 Načrtnite štruktúru rámca (frame)

Ethernet je jedným z internetových protokolov, ktorý definuje spôsob, ako posielat dáta na lokálnej sieti (tzn. Medzi počítačmi doma, alebo vo firme). Pre adresovanie (komu, od koho) používa už spomínané MAC adresy (adresy konkrétnych zariadení).

Podľa tohto protokolu sa teda dáta posielajú v Ethernetových rámcoch (Ethernet **frame**)



Rámec preto, že sú v ňom "zarámované" všetky ďalšie informácie.

b1 Popíšte spôsoby prístupu na médium, použite obrazovú prílohu č. 13

1.CSMA/CD - Používa sa pri prenosových médiách, ako je Ethernet. Zariadenie najprv počúva, či je médium voľné, a potom odosiela dáta. Ak dôjde k kolízii (viacero zariadení súčasne odosiela dáta), zariadenia detekujú kolíziu a náhodne čakajú na určitý čas pred opätovným pokusom o odoslanie.

2.CSMA/CA - Používa sa pri bezdrôtových sieťach, ako je napríklad WiFi. Zariadenie najprv počúva, či je médium voľné, a potom odosiela dáta. Na rozdiel od CSMA/CD sa však snaží predchádzať kolíziám pomocou rôznych techník, ako je napríklad náhodné oneskorenie pred odoslaním.

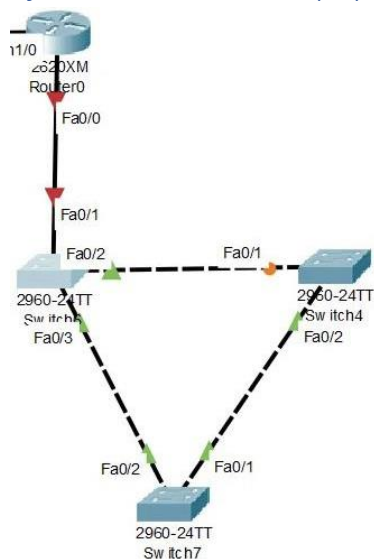
3.TDMA - Časový multiplex, ktorý prideliuje časové intervaly rôznym zariadeniam na používanie média. Každé zariadenie má pridelený časový interval, počas ktorého môže odosielať údaje.

4.FDMA - Frekvenčný multiplex, ktorý prideliuje rôznym zariadeniam rôzne frekvenčné pásma na používanie. Každé zariadenie má pridelené jedno alebo viac frekvenčných pásiem, počas ktorých môže komunikovať.

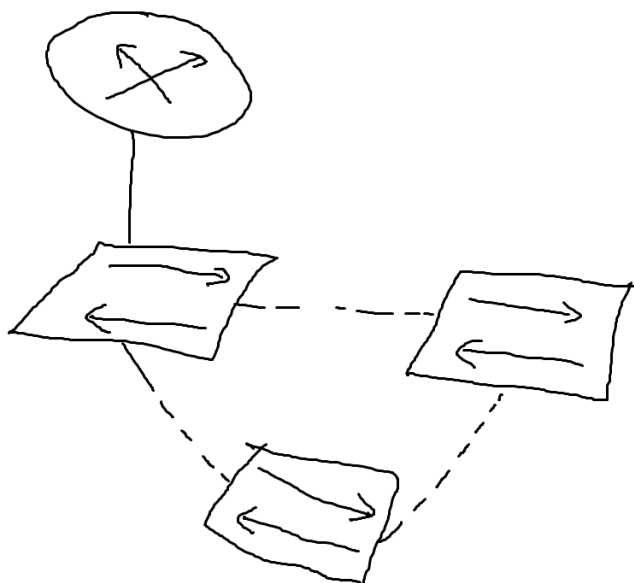
5.CDMA - Používa kódy na oddelenie údajov od rôznych zariadení. Každé zariadenie používa unikátny kód, ktorý umožňuje oddeľovať a identifikovať jeho údaje od ostatných na médium.

Tieto spôsoby prístupu na médium sa používajú na riadenie prístupu zariadení k zdieľaným prenosovým médiám a zabezpečenie efektívneho a spoľahlivého prenosu údajov v sieťach.

b2 Načrtnite schému zapojenia troch switchov pripojených k smerovaču (routru)



// Peťo: takto to asi na tabuli nenačrtneš. Takže som spravil niečo takéto:



b3 Charakterizujte cyklický vývoj trhovej ekonomiky

Cyklický vývoj trhovej ekonomiky sa odkazuje na pravidelné zmeny v ekonomickej aktivite, ktoré nasledujú za sebou v určitých obdobiach. Charakterizuje sa obvykle ako cyklus ekonomickej expanzie, vrcholu, recesie a dna, pričom sa opakuje v určitých intervaloch. Tu sú hlavné charakteristiky jednotlivých fáz cyklu:

- 1. Ekonomická expanzia** - Po období recesie alebo útlmu nasleduje obdobie expanzie. V tejto fáze ekonomika rastie, dochádza k nárastu HDP, investícií, spotreby a zamestnanosti. Firmy investujú do nových projektov, spotrebitelia majú vyššiu dôveru a rastie celková produktivita.
- 2. Vrchol** - Po dosiahnutí vrcholu expanzie nasleduje obdobie, kedy ekonomický rast dosahuje svoj maximálny bod. Rast sa spomaľuje, investície môžu dosiahnuť stav pretečenia a inflácia môže začať rásť. Rastú obavy o prehrievanie ekonomiky a infláciu.
- 3. Recesia** - Po vrchole nasleduje recesia, ktorá je charakterizovaná poklesom ekonomickej aktivity. Dochádza k poklesu HDP, investícií, spotreby a zamestnanosti. Firmy môžu znížiť svoje investície a prepúšťať zamestnancov, čo vedie k poklesu dôvery spotrebiteľov a zníženiu spotreby.
- 4. Dno** - Recesia sa môže stabilizovať na určitom bode a začína sa fáza dna, kedy ekonomika dosahuje svoje najnižšie body. V tomto bode sú firmy často najviac zraniteľné, ale súčasne sa začínajú objavovať prvé známky oživenia.

c1 Zhodnoťte dôsledky cyklického vývoja ekonomiky (inflácia, nezamestnanosť)

Cyklický vývoj ekonomiky má vplyv na rôzne ekonomické ukazovatele, vrátane inflácie a nezamestnanosti.

1. Inflácia

-Počas ekonomickej expanzie môže inflácia narastať, keďže dopyt po tovare a pracovnej sile presahuje kapacitu ekonomiky. Firmy zvyšujú ceny v reakcii na silný dopyt a nedostatok zdrojov, čo vedie k inflačným tlakom.

-Po vrchole a vstupe do recesie sa rast inflácie obvykle spomaľuje alebo stabilizuje. Nižšia spotreba a nižšie investície v dôsledku poklesu dopytu môžu znížiť inflačné tlaky.

2. Nezamestnanosť

-Počas ekonomickej expanzie sa obvykle znižuje nezamestnanosť, keďže firmy zvyšujú investície a zamestnávajú nových pracovníkov na zvládnutie rastúcej produkcie.

-Po dosiahnutí vrcholu a vstupe do recesie sa nezamestnanosť zvyčajne zvyšuje. Firmy môžu znížiť počet zamestnancov v reakcii na pokles dopytu a ziskov, čo vedie k rastúcej nezamestnanosti.

c2 Vysvetlite čo sú to metódy a funkcie, vytvorte vlastnú ľubovoľnú metódu a predvedte jej zavolanie

Metódy a funkcie sú základné pojmy v programovaní, ktoré slúžia na skupinovaniu príkazov do logických celkov a ich opätovné používanie.

Metóda - V objektovo orientovaných programovacích jazykoch je metóda časťou triedy a definuje akcie, ktoré možno vykonať s objektami tejto triedy. Metóda môže prijímať vstupné hodnoty (parametre) a vracaať výstupné hodnoty.

Funkcia - V niektorých programovacích jazykoch, ako je napríklad Python, funkcia je oddelená od objektov a môže byť samostatnou časťou kódu, ktorá vykonáva určité úlohy. Funkcie môžu prijímať argumenty a vracať výsledky.

```
using System;

class Program
{
    // Definícia vlastnej metódy
    static void Pozdrav(string meno)
    {
        // Táto metóda vypíše pozdrav
        // pre zadané meno
        Console.WriteLine("Ahoj, " +
            meno + "! Vitaj v mojom programe.");
    }

    static void Main(string[] args)
    {
        // Zavolanie metódy
        Pozdrav("Peter");
    }
}
```

Tento kód definuje metódu `Pozdrav`, ktorá prijíma jeden parameter `meno` a vypíše pozdrav pre toto meno. Metóda je potom zavolaná v `Main` metóde s argumentom "Peter", čo vypíše správu "Ahoj, Peter! Vitaj v mojom programe."

14. PCY, OSY – Architektúra počítača a operačné systémy – Dominik Hamino

a)

1. Špecifikujte základné komponenty personálneho počítača strednej triedy
2. Vysvetlite úlohy operačného systému
3. Porovnajte blokovú schému von Neumanovej a Harvardskej architektúry počítača, použite obrazovú prílohu č. 14
4. Definujte formáty základných dosiek s popisom čipovej sady severného a južného mostíka
5. Charakterizujte typy čipov ROM BIOSu

b)

1. Vymenujte a popíšte spôsob použitia niektorých interných pamätí v počítači
2. Vysvetlite vývoj pamäťových modulov RAM v počítači
3. Charakterizujte vývoj a druhy operačných systémov

c)

1. Vysvetlite hlavné úlohy personálneho manažmentu v podniku
2. Charakterizujte objektovo orientované programovanie a vysvetlite základné pojmy: trieda, objekt, členy tried atď.

a1 Špecifikujte základné komponenty personálneho počítača strednej triedy

- 1. Procesor (CPU):** Srdce počítača, ktoré vykonáva všetky výpočty a operácie. V strednej triede to môže byť viacjadrový procesor s dostatočným výkonom na bežné úlohy ako je práca s dokumentmi, surfovanie po internete alebo hranie hier.
- 2. Operačná pamäť (RAM):** Pamäť používaná na okamžité ukladanie a získavanie údajov pre aktuálne spustené programy a procesy. Počítače strednej triedy môžu mať obvykle 8 GB až 16 GB RAM.
- 3. Úložný priestor (pevný disk alebo SSD):** Slúži na trvalé ukladanie dát, ako sú aplikácie, dokumenty, média atď. Pevné disky s kapacitou 500 GB až 2 TB alebo SSD s podobnými kapacitami sú bežné pre strednú triedu.
- 4. Grafická karta (GPU):** Je zodpovedná za vykresľovanie obrázkov na obrazovke. V počítačoch strednej triedy môže byť integrovaná grafická karta, ktorá zdieľa pamäť s hlavným systémom alebo diskretná grafická karta, ktorá má vlastnú pamäť.
- 5. Základná doska (Motherboard):** Komponent, ktorý spája všetky ostatné komponenty dohromady a umožňuje im komunikovať medzi sebou.
- 6. Napájanie (Power Supply Unit - PSU):** Zdroj energie, ktorý dodáva elektrický prúd do všetkých komponentov počítača.
- 7. Chladič a ventilátor(y):** Chladič udržiava teplotu procesora a iných dôležitých komponentov na bezpečnej úrovni. Ventilátory pomáhajú pri odvádzaní tepla z počítača.
- 8. Zvuková karta:** Riadi zvukový výstup a vstup pre reproduktory, slúchadlá, mikrofóny a iné zvukové zariadenia.
- 9. Sieťová karta (Ethernet alebo Wi-Fi):** Umožňuje pripojenie k internetu a lokálnej sieti.

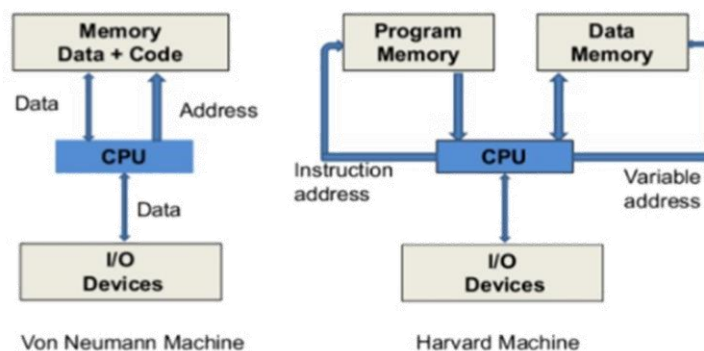
a2 Vysvetlite úlohy operačného systému

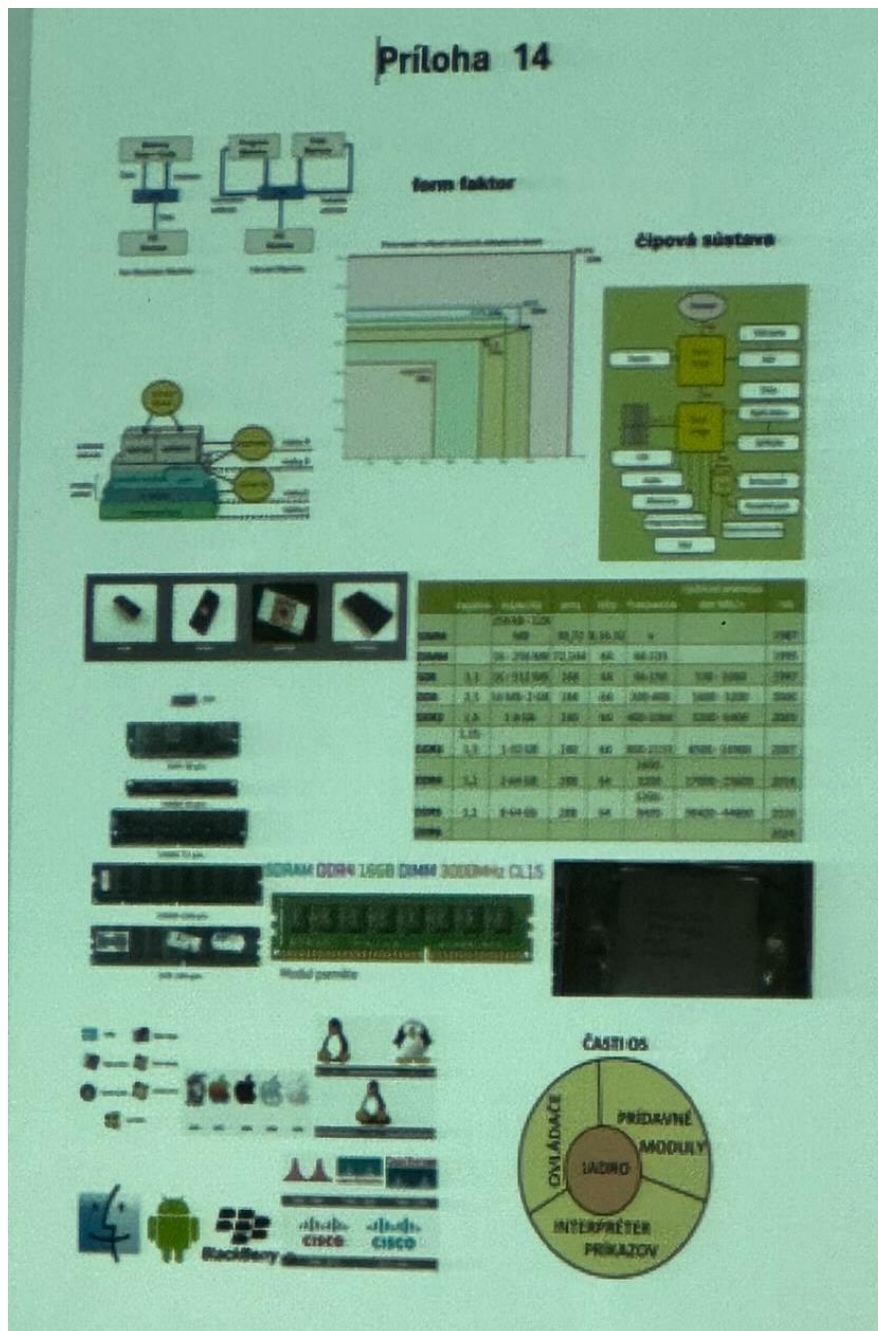
- 1. Správa zdrojov:** Operačný systém riadi prístup k hardvérovým zdrojom, ako sú procesor, pamäť, úložný priestor a periférne zariadenia. Zabezpečuje, aby rôzne programy a procesy mohli zdieľať zdroje efektívne a bezpečne.
- 2. Správa súborov:** OS zabezpečuje organizáciu a správu súborov na úložných zariadeniach. To zahŕňa vytváranie, premenu, kopírovanie, premiestňovanie a mazanie súborov a priečinkov. Taktiež zabezpečuje, aby boli súbory chránené pred neoprávneným prístupom a stratením.
- 3. Prostredie užívateľa:** Poskytuje užívateľské rozhranie, ktoré umožňuje interakciu s počítačom. To môže zahŕňať grafické používateľské rozhranie (GUI), príkazový riadok alebo iné formy interakcie.
- 4. Správa procesov:** Operačný systém riadi spustené programy a procesy, vrátane alokácie procesora, časových plánov, riadenia pamäti a synchronizácie medzi procesmi.
- 5. Komunikácia so zariadeniami:** Zabezpečuje komunikáciu s periférnymi zariadeniami, ako sú klávesnice, myši, tlačiarne, monitor, sieťové adaptéry a ďalšie.
- 6. Zabezpečenie:** Operačný systém zabezpečuje ochranu pred neoprávneným prístupom, škodlivým softvérom a ďalšími bezpečnostnými hrozbami. To môže zahŕňať prístupovú kontrolu, šifrovanie údajov, firewall a ďalšie bezpečnostné opatrenia.
- 7. Správa sietí:** V prípade operačných systémov siete zabezpečuje komunikáciu a správu v rámci lokálnej siete a pripojenie k internetu.

a3 Porovnajete blokovú schému von Neumanovej a Harvardskej architektúry počítača, použite obrazovú prílohu č. 14

Harvardská architektúra bola založená na pôvodnom počítači založenom na relé Harvard Mark I 1937, pôvodná harvardská architektúra ukladala pokyny na diernu pásku a údaje do elektromechanických počítačiel.

Von Neumannova architektúra tvorí základ moderných výpočtových technológií a je ľahšie implementovateľná využíva sa v stolových počítačoch, notebookoch, výkonných počítačoch až po pracovné stanice. V súčasnosti sa najčastejšie stretávame s počítačmi, ktoré majú von Neumannovu architektúru, až na malé výnimky je jeho schéma platná dodnes. Ide o zmenu analógového a digitálneho signálu.





a4 Definujte formáty základných dosiek s popisom čipovej sady severného a južného mostíka

1. ATX (Advanced Technology eXtended):

Popis: ATX je najbežnejší formát základných dosiek pre stolové počítače. Majú štandardizované rozmery a montážne otvory, ktoré umožňujú kompatibilitu s väčšinou počítačových skríň.

Severný mostík (Northbridge): V moderných základných doskách je funkcie severného mostíka integrovaná priamo do procesora (CPU).

Južný mostík (Southbridge): Zvyčajne obsahuje periférne rozhrania ako USB, SATA, PCI a ďalšie.

2. Micro-ATX (mATX):

Popis: mATX je menší formát základných dosiek, ktorý ponúka väčšinu funkcií plnohodnotnej ATX dosky, ale s menšími rozmermi.

Severný mostík (Northbridge): Podobne ako u ATX dosiek, moderné procesory integrujú funkcie severného mostíka.

Južný mostík (Southbridge): Má podobné funkcie ako u ATX dosiek.

3. Mini-ITX:

Popis: Mini-ITX je najmenší bežný formát základných dosiek, čo z neho robí ideálnu voľbu pre malé počítačové zostavy a HTPC (Home Theater PC).

Severný mostík (Northbridge): Integrovaný do procesora.

Južný mostík (Southbridge): Tiež má integrované periférne rozhrania.

a5 Charakterizujte typy čipov ROM BIOSu

ROM - pamäť len na čítanie

Nevyhnutným počítačovým čipom je čip s pamäťou iba na čítanie.

Čipy ROM sú umiestnené na základnej doske a na ďalších doskách s plošnými spojmi a obsahujú inštrukcie, ku ktorým má CPU priamy prístup.

ROM je energeticky nezávislý, čo znamená, že pri vypnutí počítača sa jeho obsah nevymaže.

P = programming

E = erasable = vymazateľné ultrafialovým svetlom

EE = electrically erasable

b1 Vymenujte a popíšte spôsob použitia niektorých interných pamätí v počítači

Operačná pamäť (RAM) je pamäť v ktorej je uložený prebiehajúci program a údaje s ktorými spolupracuje procesor.

Registre procesora (RAM) je pamäťové miesto, ktoré slúži procesoru na uchovávanie údajov, ktoré práve spracováva. Ide o pomerne malé množstvo veľmi rýchlej pamäte, ktorá je priamo súčasťou procesorového jadra a prístup k nemu závisí od inštrukčnej sady.

vyrovnávacia pamäť (=buffer) (RAM)

b2 Vysvetlite vývoj pamäťových modulov RAM v počítači

DRAM (Dynamic RAM): Začiatky pamäťových modulov RAM siahajú do obdobia 70. a 80. rokov, kedy sa používala DRAM. Tieto moduly vyžadovali pravidelné obnovovanie údajov, inak by sa stratili.

Kapacita jednotlivých pamäťových čipov bola pomerne nízka a dosky mohli obsahovať len niekoľko megabajtov pamäte.

SDRAM (Synchronous DRAM): S nástupom SDRAM v 90. rokoch sa zvýšila rýchlosť pamäte v porovnaní s tradičnými DRAM modulmi. Komunikácia medzi procesorom a pamäťou bola synchronizovaná, čo zlepšilo celkový výkon.

SDRAM bol rozšírený a používal sa v počítačoch počas väčšiny 90. a začiatku 2000. rokov.

DDR (Double Data Rate SDRAM): DDR pamäťové moduly, ktoré sa objavili v polovici 2000. rokov, priniesli ďalšie zvýšenie rýchlosti. Tieto moduly mohli prenášať dáta dvakrát počas každého hodinového cyklu, čo znamenalo väčšiu priepustnosť údajov.

DDR pamäť sa ďalej vyvíjala až po sériu ako DDR4, poskytujúca vyššiu rýchlosť a nižšiu spotrebu energie.

DDR2, DDR3, DDR4 a DDR5: Každá nasledujúca generácia DDR pamäte priniesla zvyšujúcu sa rýchlosť, nižšiu spotrebu energie a vyššiu kapacitu.

DDR3 pamäť bola dominantná v počítačoch od začiatku 21. storočia do prvej polovice 2010. rokov, kedy bola nahradená DDR4 pamäťou.

DDR4 pamäť priniesla ešte vyššiu rýchlosť a nižšiu spotrebu energie, a je štandardom v mnohých moderných počítačoch.

V roku 2020 a neskôr sa začala uvádzať pamäť **DDR5**, ktorá prináša ďalšie zvýšenie rýchlosti a efektivity v porovnaní s predchádzajúcimi generáciami.

b3 Charakterizujte vývoj a druhy operačných systémov

Skoré operačné systémy: V skorých dňoch počítačových vedeckých a výskumných laboratórií, v 50. a 60. rokoch 20. storočia, boli operačné systémy základné a veľmi primitívne. Jedným z najznámejších skorých operačných systémov bol UNIX, vyvinutý v laboratóriách Bell Labs v 70. rokoch.

Grafické užívateľské rozhrania (GUI): S nástupom osobných počítačov a návratom spoločnosti Apple na trh s počítačmi s uvedením Macintosh v roku 1984 a neskôr s príchodom Microsoft Windows 3.0 v roku 1990 sa začali objavovať operačné systémy s grafickými užívateľskými rozhraniami

Multitasking a multiprocessorové systémy: S nástupom silnejších hardvérových technológií sa operačné systémy začali vyvíjať s podporou multitaskingu a multiprocessingu. To znamená, že môžu bežať viacero úloh súčasne a využívať viacero procesorov.

Mobilné operačné systémy: S nástupom mobilných zariadení ako smartfóny a tablety, vznikli nové operačné systémy zamerané na mobilné platformy. Medzi najznámejšie patria iOS od Apple pre iPhone a iPad, Android vyvinutý spoločnosťou Google a Windows Phone od spoločnosti Microsoft (konzumácia bola ukončená v roku 2017).

Cloud operačné systémy: S nástupom cloudových výpočtových technológií vznikajú nové operačné systémy navrhnuté pre prácu v cloude. Tieto systémy sa bežne nachádzajú na serveroch a umožňujú prístup k aplikáciám a údajom prostredníctvom internetu. Príklady zahŕňajú Google Chrome OS a Amazon Web Services.

c1 Vysvetlite hlavné úlohy personálneho manažmentu v podniku

1. Personálne plánovanie - odhady a prepočty potreby zamestnancov, plánovanie personálneho rozvoja zamestnancov

2. Analýza práce - vymedzenie pracovných miest a funkcií, spracovanie opisov práce a špecifikácii požiadaviek na zamestnancov

3. Získanie a výber zamestnancov - zabezpečenie obsadenia voľných prac. Miest

- 4. Rozmiestňovanie a kariéra zamestnancov** - prijímanie nových zamestnancov, povýšenie, premiestnenie na nižšiu funkciu, preradenie na inú prácu skončenie prac. pomeru
- 5. Kvalifikačná príprava a kariéra zamestnancov** - zabezpečenie nepretržitého vzdelávani
- 6. Hodnotenie zamestnancov** - vypracovanie a uplatňovanie systému hodnotenia zamestnancov
- 7. Motivácia a stimulácia zamestnancov**
- 8. Odmeňovanie zamestnancov** - mzda a poskytovanie rôznych výhod
- 9. Pracovné vzťahy** - medzi vedením podniku a odbormi, komunikácia v podniku, zlepšovanie vzťahov medzi zamestnancami
- 10. Pracovné podmienky** - časové podmienky práce, pracovné prostredie, bezpečnosť a ochrana pri práci
- 11. Personálny a informačný systém** - evidovanie a spracovanie údajov o zamestnancoch, pracovných miestach, mzdách, sociálnych záležitostiach

c2 Charakterizujte objektovo orientované programovanie a vysvetlite základné pojmy: trieda, objekt, členy tried atď.

Objektovo orientované programovanie (OOP) je paradigma programovania, ktorá sa zameriava na modelovanie skutočných svetových entít pomocou objektov a interakcií medzi nimi.

Trieda: Trieda je základným stavebným kameňom OOP. Je to abstraktný model, ktorý definuje vlastnosti a správanie objektov. Trieda obsahuje atribúty (premenné) a metódy (funkcie alebo procedúry), ktoré popisujú vlastnosti a operácie, ktoré môže objekt vykonávať.

Objekt: Objekt je konkrétna inštancia triedy. Znamená to, že objekt je reálna entita, ktorá existuje v pamäti počítača a má svoje vlastné atribúty a môže vykonávať operácie definované v triede, ktorá mu zodpovedá. Objekt je "skutočnou" realizáciou triedy.

Členy triedy: Členy triedy sú vlastnosti a metódy, ktoré sú definované v triede.

Atribúty (premenné): Atribúty triedy sú premenné, ktoré uchovávajú dáta pre každý konkrétny objekt. Môžu reprezentovať stav alebo vlastnosti objektu.

Metódy (funkcie/procedúry): Metódy triedy sú funkcie alebo procedúry, ktoré operujú nad dátami objektu. Môžu meniť stav objektu alebo vykonávať rôzne operácie.

Dedičnosť: Dedičnosť je proces, ktorý umožňuje vytvárať nové triedy (tzv. podtriedy alebo potomkov) na základe existujúcej triedy (tzv. nadtriedy alebo rodiča). Potomky zdedia vlastnosti a metódy nadtriedy a môžu definovať ďalšie špecifické vlastnosti alebo správanie.

15. Programovanie – Algoritmy a ich vlastnosti, vývojové diagramy – Martin Mikuláš

a)

1. Uveďte, čo rozumieme pod algoritmom v procese programovania

2. Uvedte, čo je to vývojový diagram
3. Charakterizujte základné vlastnosti algoritmov, uveďte konkrétne príklady
4. Ak si uvedomíme, že program je vlastne algoritmus prepísaný do konkrétneho programovacieho jazyka, uveďte pri nesplnení ktorých vlastností je možné takýto program skompilovať a spustiť
5. Uveďte a popíšte základné zložky algoritmov

b)

1. Popíšte etapy algoritmizácie
2. Nakreslite základné značky vývojového diagramu a vysvetlite ich
3. Navrhните algoritmus a vývojový diagram na určenie väčšieho čísla z dvoch celých čísel zadaných z klávesnice

c)

1. Vysvetlite pojem redundancia na L2 (STP, Etherchannel, ...)
2. Charakterizujte mzdu ako cenu práce na trhu a na konkrétnom príklade uveďte výpočet čistej mzdy zamestnanca v podniku

a1 Uvedte, čo rozumieme pod algoritmom v procese programovania

Algoritmus je postup, pravidlo ktorý sa používa na vykonanie určitej úlohy. Je to postupnosť krokov, ktoré vedú k nejakému vyriešeniu problémov. Napríklad raňajky. Najskôr si vyberiem chlieb, potom natrime maslo, dám šunku....

Algoritmus v programovaní je presný postup definovaných krokov, ktorá premení vstupné údaje na výstupne. Algoritmus vlastne hovorí počítaču čo má urobiť. Je to inštrukcia ktorú počítač nasleduje. Je kód ktorý sa po pridání vstupných parametrov do algoritmu vykoná.

Ide o presný návod alebo plán, ako vykonať určitú úlohu alebo dosiahnuť cieľ. Algoritmy môžu byť použité na rôzne úlohy, od triedenia čísel a vyhľadávania informácií v databáze po riadenie robotov a analýzu dát.

a2 Uvedte, čo je to vývojový diagram

Je to grafické znázornenie algoritmu. Uľahčuje to zdieľanie medzi ľuďmi, ktorí tie značky poznajú. Značky sú medzinárodné poznané. Je to vlastne plán vývoja programu.

Vývojový diagram je typ diagramu, ktorý ukazuje vzťahy medzi jednotlivými časťami procesu, postupu alebo počítačového algoritmu, ktoré sú znázornené symbolmi rôznych tvarov (štvorec, kruh...). Tie sú navzájom prepojené šípkami, ktoré znázorňujú priebeh celého procesu.

a3 Charakterizujte základné vlastnosti algoritmov, uveďte konkrétne príklady

- **Elementárnosť (jednoduchosť)** – chceme dosiahnuť čo najjednoduchšie kroky
- **Efektívnosť - časová a pamäťová:**
 - Časová - program zbehne čo najrýchlejšie, vyriešiť problém za čo najkratší možný čas
 - Pamäťová - zaberanie čo najmenej pamäti
- **Konečnosť** - skončí za rozumný čas. Postup, ktorý vždy skončí po vykonaní konečného počtu krokov .

- **Hromadnosť** – da sa použiť na viac príkladov. Napr. program nespočíta $7 + 5$, ale všeobecne 2 čísla
- **Presnosť (Rezultatívnosť)** – výsledok má byť čo najpresnejší. Dá správny výsledok. Napr. síce sa ten program spustí, ale nevyhodí správny výsledok $5 * (-6) = 30$ výsledok má byť -30
- **Determinovanosť** – v každom kroku algoritmu vieme aký ďalší krok nasleduje. Postup je zostavený tak, že v každom momente vykonávania je jednoznačne určené, aká činnosť má nasledovať ďalej, alebo či sa postup skončil

a4 Ak si uvedomíme, že program je vlastne algoritmus prepísaný do konkrétneho programovacieho jazyka, uvedte pri nesplnení ktorých vlastností je možné takýto program skompilovať a spustiť

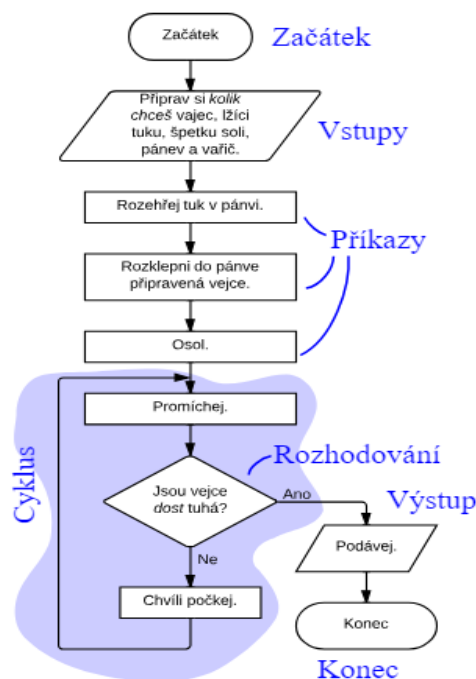
Program je možné spustiť aj vtedy, keď nespĺňa vlastnosti algoritmov. Medzi také vlastnosti patrí:

- **konečnosť**: program síce spustíme, ale on sa nikdy nedokončí, čím je porušená vlastnosť algoritmov
- **efektívnosť**: program môžeme spustiť aj keď nie je efektívny
- **hromadnosť**: program môžeme spustiť aj keď rieši len problém napr. výpočtu $3 * 3$. Taký program je síce na nič, ale kompilátor ho skompiluje a aj spustí.
- **správnosť zápisu**: vývojové prostredie nám síce nepíše žiadnu chybu a program je správne zapísaný, ale napriek tomu buď nemusí nič robiť alebo proste nebude fungovať.

Aj napriek nesplneniu týchto vlastností program skompilovať a spustiť pôjde.

a5 Uvedte a popíšte základné zložky algoritmov

- **Vstupné údaje (Input):**
 - Vstupné údaje sú informácie, na ktorých algoritmus pracuje alebo ktoré potrebuje na vykonanie svojej úlohy. Môžu to byť hodnoty, zoznamy, súbory alebo akékoľvek iné dáta.
- **Výstupné údaje (Output):**
 - Výstupné údaje sú výsledkom vykonania algoritmu. Sú to informácie alebo hodnoty, ktoré algoritmus generuje ako odpoveď na svoju úlohu alebo spracovanie vstupných údajov.
- **Operácie:**
 - Operácie sú jednotlivé kroky, ktoré algoritmus vykonáva na spracovanie vstupných údajov a generovanie výstupných údajov. Tieto kroky môžu zahŕňať aritmetické operácie, priradenia, podmienky, cykly a ďalšie.
- **Postupnosť:**
 - Postupnosť definuje poradie, v ktorom sa majú vykonávať operácie alebo akcie v algoritme. Každá akcia má svoje určené miesto v poradí, ktoré ovplyvňuje výsledok algoritmu. každý riadok sa vykoná len raz, 2 čísla, spočítaj ich, výsledok
- **Vetvenie:**
 - Vetvenie umožňuje algoritmu vykonávať rôzne kroky v závislosti od splnenia určitých podmienok. Podľa výsledku podmienky môže algoritmus zvoliť rôzne cesty vykonávania.
- **Cykly:**
 - Umožňujú opakované vykonávanie určitých krokov alebo akcií, kým je splnená určitá podmienka. (niečo v cykle sa niekoľko krát zopakuje a potom sa pokračuje ďalej v programe)



b1 Popíšte etapy algoritmizácie

• Formulácia problému:

- Na začiatku je dôležité jasne definovať problém, ktorý sa má riešiť. To zahŕňa identifikáciu vstupov a výstupov, pochopenie požiadaviek a cieľov úlohy a stanovenie kritérií pre dosiahnutie úspechu. Zadaný cieľ alebo úloha musí byť presne špecifikovaná a jasne definovaná, aby bol následný proces algoritmizácie efektívny.

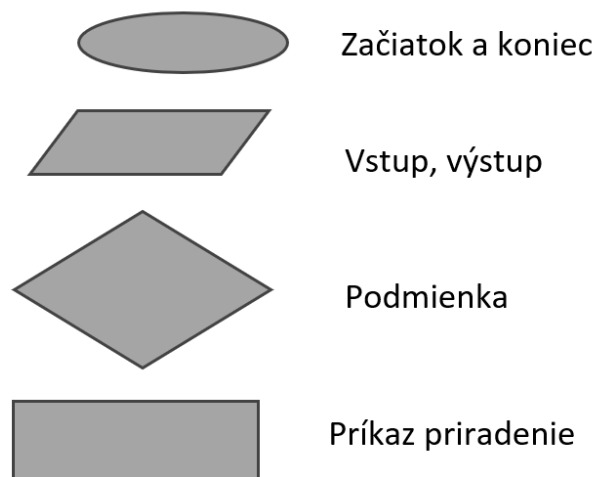
• Analýza problému:

- Po formulácii problému nasleduje dôkladná analýza, počas ktorej sa skúma štruktúra, charakteristika a podmienky problému. Táto analýza zahŕňa identifikáciu rôznych aspektov problému, potenciálnych obmedzení a možných prístupov k riešeniu. Cieľom analýzy je získať hlboké pochopenie problému a príslušných faktorov, ktoré ovplyvňujú jeho riešenie.

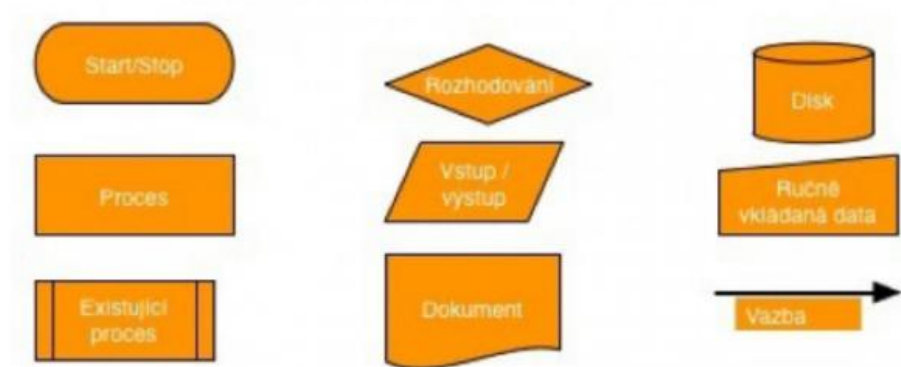
• Riešenie problému:

- Po analýze problému sa navrhne a vykonáva samotné riešenie. To zahŕňa vytvorenie algoritmu alebo postupu krokov, ktoré vedú k dosiahnutiu požadovaného výsledku. Algoritmus musí byť navrhnutý tak, aby bol efektívny, spoľahlivý a zodpovedal požiadavkám a cieľom úlohy. Po vytvorení algoritmu nasleduje jeho implementácia, testovanie a potenciálne ladenie, aby sa dosiahol požadovaný výkon a správnosť.

b2 Nakreslite základné značky vývojového diagramu a vysvetlite ich

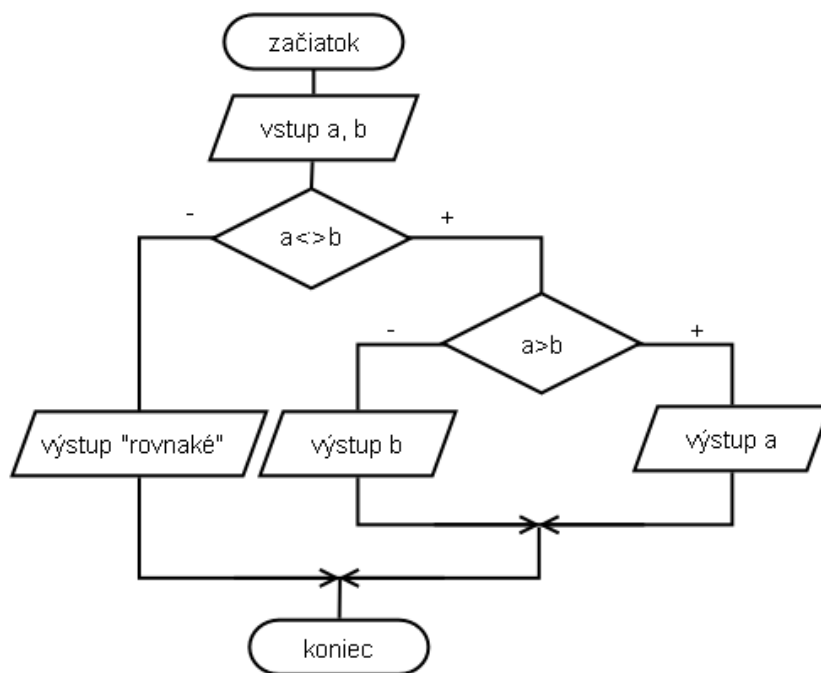


Vývojový diagram - základní symboly



b3 Navrhnete algoritmus a vývojový diagram na určenie väčšieho čísla z dvoch celých čísel zadaných z klávesnice

1. Algoritmus začne zadaním dvoch čísel **a** a **b**.
2. Skontroluje sa, či sú **a** a **b** rovnaké. Ak áno, algoritmus sa ukončí a zobrazí sa správa "rovnaké".
3. Ak **a** a **b** nie sú rovnaké, algoritmus skontroluje, či je **a** väčšie ako **b**. Ak áno, algoritmus sa ukončí a zobrazí sa číslo **b**.
4. Ak **a** nie je väčšie ako **b**, algoritmus sa ukončí a zobrazí sa číslo **a**.



c1 Vysvetlite pojem redundancia na L2 (STP, Etherchannel, ...)

Redundancia na úrovni 2 (L2) sa týka vytvorenia záložných alebo alternatívnych ciest a zdrojov v počítačovej sieti s cieľom zvýšiť spoľahlivosť a odolnosť siete voči výpadkom alebo chybám. Dva hlavné mechanizmy, ktoré sa často používajú na dosiahnutie redundancie v L2 sieti, sú Spanning Tree Protocol (STP) a EtherChannel.

Spanning Tree Protocol (STP):

STP je protokol na L2, ktorý zabezpečuje, aby v sieti nevznikli slučky. Slučky môžu vzniknúť v dôsledku existencie viacerých cestovných ciest medzi dvoma uzlami siete. STP analyzuje topológiu siete a blokuje jednu zo zbytočných ciest, čím sa zabráni vzniku slučiek. Ak dojde k výpadku alebo chybe na aktívnej ceste, STP automaticky prepojí na záložnú cestu, čo zabezpečuje redundanciu a odolnosť siete.

EtherChannel:

EtherChannel umožňuje zväzok viacerých fyzických prenosových káblov alebo rozhraní do jedného logického spojenia. Tým sa zvyšuje prenosová kapacita a zároveň sa poskytuje redundancia. Ak sa jedna z linkov v zväzku vypne alebo vynechá, EtherChannel stále poskytuje prenosovú kapacitu pomocou zvyšných linkov. Konfigurácia EtherChannel zabezpečuje, že v prípade potreby môžu byť všetky linky v zväzku využité na prenos dát.

c2 Charakterizujte mzdu ako cenu práce na trhu a na konkrétnom príklade uveďte výpočet čistej mzdy zamestnanca v podniku

NIČ

16. Programovanie – Základné vstupno-výstupné operácie a výpočty – Martin Miki

a)

1. Popíšte objekty a metódy používané pri konzolovom výstupe a vstupe
2. Uveďte, ako sú vstupy a výstupy realizované v grafických rozhraniach
3. Definujte pojmy premenná, konštanta a popíšte konverzie medzi reťazcami a číslami
4. Uveďte, na čo slúži operátor priradenia
5. Vymenujte a popíšte aritmetické operátory a ich prioritu

b)

1. Vymenujte a popíšte niektoré matematické funkcie a metódy
2. Popíšte spôsob volania matematických funkcií a metód
3. Vytvorte program na výpočet polomeru kruhu, pri jeho známom obsahu zadanom z klávesnice

c)

1. Zdôvodnite potrebu vysielania broadcast v počítačovej sieti
2. Uveďte aké náklady, výnosy a výsledok hospodárenia vznikajú v podniku (analýza nulového bodu)

a1 Popíšte objekty a metódy používané pri konzolovom výstupe a vstupe

Objekt Console je hlavným objektom používaným pri konzolovom vstupe a výstupe v C#. Poskytuje metódy na výpis textu do konzoly a na načítanie vstupu od používateľa.

Metódy používané pri konzolovom vstupe a výstupe v C#:

WriteLine(): Metóda WriteLine() slúži na výpis textu do konzoly a automaticky pridáva nový riadok na konci výstupu.

Write(): Metóda Write() slúži na výpis textu do konzoly bez pridania nového riadku na konci výstupu.

ReadLine(): Metóda ReadLine() sa používa na načítanie vstupu zo konzoly. Vráti reťazec obsahujúci celý riadok textu, ktorý zadal používateľ.

Read(): Metóda Read() sa používa na načítanie jedného znaku zo vstupu zo konzoly. Vráti kód znaku v podobe celého čísla.

Pri výstupe je tu objekt Console a metóda Write() alebo WriteLine(). Napríklad Console.WriteLine()
Metóda na vstup: Console.ReadLine();

a2 Uveďte, ako sú vstupy a výstupy realizované v grafických rozhraniach

Vstupy v grafických rozhraniach sú realizované napríklad v textBoxoch. V textboxe je to zapísané ako string. Musíme to potom konvertovať. Napríklad Convert.ToInt32()

Výstupy v grafických rozhraniach v listBox alebo label

- **Vstupy v grafických rozhraniach v C#:**

- TextBox je používaný pre vstup textu od používateľa. Umožňuje používateľovi písať alebo editovať text.
- Button je interaktívny prvok, ktorý umožňuje používateľovi vykonať určitú akciu kliknutím.
- ComboBox je kombinovaný prvok, ktorý kombinuje textové pole s rozbaľovacím zoznamom možností, z ktorých používateľ môže vybrať.
- RadioButton je používaný v prípadoch, keď používateľ má vybrať jednu z viacerých možností.
- CheckBox umožňuje používateľovi označiť alebo odznačiť jednu alebo viacero možností.

- **Výstupy v grafických rozhraniach v C#:**

- Label slúži na zobrazenie textu alebo popisu v GUI. Používa sa na zobrazenie statického textu alebo na označenie iných prvkov v rozhraní.
- PictureBox sa používa na zobrazenie obrázkov alebo grafických prvkov v GUI.
- ListBox: ListBox slúži na zobrazenie zoznamu položiek, z ktorých používateľ môže vybrať. Používa sa na zobrazenie dynamicky naplnených zoznamov, ako sú napríklad zoznamy možností alebo výsledkov vyhľadávania.

ListView alebo DataGridView sa používajú na zobrazenie tabuľkových dát v GUI.

a3 Definujte pojmy premenná, konštanta a popíšte konverzie medzi reťazcami a číslami

Premenná je označenie v pamäti počítača, ktoré je priradené určitým dátam a umožňuje prístup k týmto dátam. Každá premenná má svoj jedinečný názov a dátový typ, ktorý definuje typ údajov, ktoré môže táto premenná obsahovať. // pečo: Premenná int má 4 byty. Premenná môže byť vytvorená, inicializovaná a následne používaná v rôznych častiach programu na uchovávanie a manipuláciu s dátami. Celočíselná je int, premenná na desatinné čísla je double, float

Premenná preto, lebo môžem počas používania programu zmeniť jej hodnotu. Premennú najskôr deklarujeme int cislo

Konštanta – nemenná hodnota, konštantne daná. Konštanta je hodnota, ktorá je definovaná v programe a nemôže sa počas behu programu zmeniť. Na rozdiel od premenných, konštanty majú pevnú hodnotu, ktorá je určená pri deklarácii a nemôže byť zmenená. Konštanty sú často používané na uloženie hodnôt, ktoré sa počas behu programu nemôžu meniť, ako napríklad matematické konštanty alebo hodnoty konfigurácií.

Konverzia je vlastne proces premeny jedného typu dát na iný. Napríklad ako:

Prevod reťazca na číslo:

```
string strNumber = "123";
int number = Convert.ToInt32(strNumber);
```

Prevod čísla na reťazec:

```
int number = 123;
string strNumber = number.ToString();
```

a4 Uved'te, na čo slúži operátor priradenia

Operátor priradenia slúži na priradenie hodnoty do premennej. Môže byť použitý na akýkoľvek dátový typ s jedinou podmienkou, že prvý parameter (ľavá strana), musí byť prijímač hodnoty, napríklad premenná. V jazykoch programovania sa tento operátor väčšinou reprezentuje znakom "="

Napríklad: `int x = 5;`

a5 Vymenujte a popíšte aritmetické operátory a ich prioritu

Slúžia na operáciu s dvomi alebo viacerými premennými. Máme napríklad sčítanie, odčítanie, násobenie, delenie. Platí tu priorita pri násobení, ktoré sa vykoná najskôr.

Unárne: +, ++, -

```
int a = +5;
```

```
int b = -10;
```

```
int x = 5;  
x++; // x r
```

Binárne: +, -, *, /, % modulo, lebo potrebujú 2 operandy (c1, c2)

```
int result = 10 - 3;
```

```
int result = 5 + 3;
```

```
int result = 4 * 5;
```

```
double result = 10.0 / 3.0;
```

b1 Vymenujte a popíšte niektoré matematické funkcie a metódy

V programovaní sa používajú rôzne matematické funkcie a metódy. Sínus, kosínus, odmocnina, umocnenie a podobne.

V C# existuje **trieda** s názvom **Math**, ktorá poskytuje rôzne metódy napríklad Math.Sin(), Math.Cos(), Math.Sqrt() a Math.Pow()

V C++ je to funkcia, nie ako metóda triedy.

- **Sínus (sin):**
 - Math.Sin(): Vracia sínus daného uhla (v radiánoch).
 - Napríklad: Math.Sin(Math.PI/2) vráti hodnotu 1.
 - V C++: sin(angle)
- **Kosínus (cos):**
 - Math.Cos(): Vracia kosínus daného uhla (v radiánoch).
 - Napríklad: Math.Cos(0) vráti hodnotu 1.
 - V C++: cos(angle)
- **Tangens (tan):**
 - Math.Tan(): Vracia tangens daného uhla (v radiánoch).
 - Napríklad: Math.Tan(Math.PI/4) vráti hodnotu 1.
- **Odmocnina (sqrt):**
 - Math.Sqrt(): Vracia odmocninu z daného čísla.
 - Napríklad: Math.Sqrt(9) vráti hodnotu 3.

V C++: sqrt(number)

b2 Popíšte spôsob volania matematických funkcií a metód

V C# sa matematické funkcie volajú pomocou Math. Math je názov triedy. Výsledok musíme niekam priradiť

Math.Sqrt() - druhá odmocnina

Math.Round() - zaokrúhľovanie

Math.Sin()

Math.Cos()

b3 Vytvorte program na výpočet polomeru kruhu, pri jeho známom obsahu zadanom z klávesnice

```
using System;

class Program
{
    static void Main()
    {
        Console.WriteLine("Zadajte obsah kruhu:");
        double obsahKruhu = Convert.ToDouble(Console.ReadLine());

        double polomer = Math.Sqrt(obsahKruhu / (Math.PI)); // Vypočítame polomer

        Console.WriteLine("Polomer kruhu je: " + polomer);
        Console.ReadLine();
    }
}
```

c1 Zdôvodnite potrebu vysielania broadcast v počítačovej sieti

Broadcast ("všesmerové vysielanie")- údaje sa posielajú na **všetky** počítače v dosahu (obvykle sa jedná o lokálnu sieť);

- **Nájdenie zariadení v sieti:** Broadcastové správy sa často používajú na hľadanie a identifikáciu zariadení v sieti. Napríklad, keď nové zariadenie vstúpi do siete, môže odoslať broadcastovú správu, aby sa predstavilo ostatným zariadeniam.
- **Rozširovanie informácií:** V niektorých prípadoch je potrebné zdieľať dôležité informácie so všetkými zariadeniami v sieti. Broadcastové správy umožňujú jednoducho rozšíriť tieto informácie na všetky zariadenia naraz.
- **Aktualizácie a upozornenia:** Keď sa v sieti vyskytne dôležitá udalosť alebo aktualizácia, broadcastové správy môžu byť použité na oznámenie všetkým zariadeniam. Toto je užitočné pre bezpečnostné upozornenia, softvérové aktualizácie a podobné udalosti.

Dohľad a diagnostika: Administrátori sietí často využívajú broadcastové správy na monitorovanie a diagnostiku siete. Tieto správy môžu obsahovať informácie o stave siete, prevádzkovaných službách a podobne

c2 Uved'te aké náklady, výnosy a výsledok hospodárenia vznikajú v podniku (analýza nulového bodu)

Nulový bod – bod, kde nemáme stratu, ani zisk. Služi nám na plánovanie zisku.

$$NB = \frac{\text{celkové fixné náklady}}{\text{cena 1 výrobku} - \text{variabilné náklady na 1 výrobok}}$$

17. Programovanie – Riadiace príkazy-podmienené príkazy, prepínač, cykly – Zawo

a)

1. Uveďte zoznam relačných a logických operátorov
2. Popíšte vytváranie a vyhodnocovanie podmienených výrazov
3. Charakterizujte podmienený príkaz – na čo slúži, aký je jeho tvar a ako je realizovaný
4. Charakterizujte prepínač- na čo slúži, aký je jeho tvar a ako je realizovaný
5. Uveďte, v akých prípadoch sa používajú cykly

b)

1. Popíšte jednotlivé typy cyklov a uveďte, na čo je daný typ cyklu vhodný
2. Uveďte, akým spôsobom možno predčasne ukončiť cyklus a ako aktuálnu iteráciu
3. Vytvorte program na nájdenie najväčšieho čísla z N čísel, ktoré sú zadávané z klávesnice

c)

1. Popíšte funkciu multilayer switch v počítačovej sieti
2. Vysvetlite aká je úloha štátu v trhovej ekonomike

a1 Uveďte zoznam relačných a logických operátorov

Relačné operátory

== (rovná sa)

!= (nerovná sa)

> (väčší ako)

< (menší ako)

>= (väčší ako alebo rovný)

<= (menší ako alebo rovný)

Logické operátory

&& (logický AND) - Výsledok je true, ak sú oba vstupy true.

|| (logický OR) - Výsledok je true, ak aspoň jeden z vstupov je true.

! (logický NOT) - Neguje boolean hodnotu; z true robí false a naopak.

a2 Popíšte vytváranie a vyhodnocovanie podmienených výrazov

Vytváranie a vyhodnocovanie podmienených výrazov v jazyku C# sa často uskutočňuje pomocou if, else if a else.

```
if(podmienka) { výrok }
```

```
if (podmienka) { výrok } else { výrok 2 }
```

a3 Charakterizujte podmienený príkaz – na čo slúži, aký je jeho tvar a ako je realizovaný

Pomocou podmienok môžeme ovládať náš program. Podmienený príkaz môže riadiť beh programu v dvoch smeroch. Ak je podmienka splnená program bude pokračovať v prvom smere, ak nie tak v druhom smer.

Ak nevyhovuje podmienka if, dáme tam else.

if, else if, else

Napríklad `if(cislo % 5 == 0) { výrok } else { výrok 2 }`

a4 Charakterizujte prepínač- na čo slúži, aký je jeho tvar a ako je realizovaný

V prepínači v jazyku C# je použitie slova break povinné pre každý prípad (case).

```
switch (hodnota)
```

```
{
```

```
    case hodnota1:
```

```
        break;
```

```
}
```

V prepínači v jazyku C++ je použitie break kľúčového slova voliteľné, ak chceme ukončiť každý prípad (case). Ak break nie je použité, program pokračuje do nasledujúceho prípadu bez ohľadu na splnenie podmienky.

a5 Uvedte, v akých prípadoch sa používajú cykly

Cykly umožňujú opakované vykonávanie určitého bloku kódu viac krát.

b1 Popíšte jednotlivé typy cyklov a uvedte, na čo je daný typ cyklu vhodný

For(premenná; podmienka; príkaz)

Slúži na opakované vykonávanie kódu určitý počet krát

Ak je podmienka true tento cyklus sa vykoná

Presný počet opakovaní cyklu

While

Používa sa na opakované vykonávanie kódu, pokiaľ platí určitá

podmienka. Najprv sa vyhodnotí podmienka. Ak je true, vykoná sa

tento cyklus. Ak sa podmienka stane false cyklus sa ukončí

Do-while

Je podobný ako while, ale vykoná sa aspoň raz bez ohľadu na to či je podmienka True alebo false

foreach

Príkaz foreach vyčíslí prvky kolekcie a spustí jeho telo pre každý prvok kolekcie.

b2 Uvedte, akým spôsobom možno predčasne ukončiť cyklus a ako aktuálnu iteráciu. Kľúčové slovo **break** v C# umožňuje programátorovi okamžite zastaviť vykonávanie cyklu a prejsť na príkaz nasledujúci po slučke v kóde.

Na druhej strane kľúčové slovo **continue** umožňuje programátorovi preskočiť zvyšok aktuálneho prechodu slučky a pokračovať v ďalšom prechode bez vykonania kódu uvedeného nižšie.

b3 Vytvorte program na nájdenie najväčšieho čísla z N čísel, ktoré sú zadávané z klávesnice

```
Console.WriteLine("Zadajte počet čísel:");
int pocet = Convert.ToInt32(Console.ReadLine());
int max = int.MinValue; // Na začiatku nastavíme max na najmenšiu možnú hodnotu int

for (int i = 0; i < pocet; i++)
{
    Console.WriteLine("Zadajte " + i + "." + " číslo:");
    int cislo = Convert.ToInt32(Console.ReadLine());

    if (cislo > max)
    {
        max = cislo; // Ak je zadané číslo väčšie ako aktuálny max, aktualizujeme max
    }
}

Console.WriteLine("Najväčšie zadané číslo je: " + max);
Console.ReadLine();
```

c1 Popíšte funkciu multilayer switch v počítačovej sieti

Multilayer switch, alebo viacvrstvový prepínač, je zariadenie v počítačovej sieti, ktorý na rozdiel od obyčajného switcha, ktorý pracuje len na druhej vrstve OSI modelu (t.j. na úrovni linkovej vrstvy a zaoberá sa MAC adresami), multilayer switch dokáže pracovať aj na tretej vrstve, čo je sieťová vrstva. To znamená, že vie čítať IP adresy a rozhodovať, kam má dáta poslať nielen na základe MAC adresy, ale aj IP adresy. Multilayer switch je schopný nielen preposielať dáta medzi zariadeniami vo vnútri jednej lokálnej siete, ale môže tiež smerovať dáta medzi rôznymi segmentmi siete alebo dokonca rôznymi podsietami. Takto môže zvládať úlohy, ktoré by inak vyžadovali kombináciu switchu a routeru. Použitie multilayer switchu môže zefektívniť sieťovú infraštruktúru vďaka rýchlejšiemu preposielaniu dát a lepšiemu riadeniu dátovej premávky, najmä v väčších a komplexnejších sieťach. Okrem základného preposielania dát môže obsahovať aj pokročilé funkcie ako sú VLANy (virtuálne lokálne siete), Quality of Service (QoS) pre prioritizáciu dátových tokov, a rôzne bezpečnostné funkcie.

18. Programovanie – Jednoduché a zložené údajové typy – Zdenko

a)

1. Vymenujte aspoň 5 jednoduchých údajových typov, uveďte na čo slúžia a prečo sa zaraďujú medzi jednoduché údajové typy
2. Charakterizujte reťazce, predved'te vytvorenie reťazca, vstup údajov do reťazca a volanie základných metód pre prácu s reťazcami
3. Charakterizujte klasické pole, popíšte deklaráciu a vytvorenie poľa aj z hľadiska pamäte
4. Charakterizujte triedu ako údajovú štruktúru
5. Charakterizujte kontajnery a kolekcie ako úložiská údajov

b)

1. Uveďte výhody používania kontajnerov a kolekcii oproti klasickým poliam
2. Predved'te použitie základných metód pri práci s kolekciami
3. Vytvorte program, ktorý naplní klasické N prvkové pole náhodnými číslami, pričom interval generovania nech je zadaný z klávesnice

c)

1. Načrtnite topológiu s nutnosťou použitia Default Gateway
2. Popíšte základnú štruktúru podnikateľského plánu

a1 Vymenujte aspoň 5 jednoduchých údajových typov, uveďte na čo slúžia a prečo sa zaraďujú medzi jednoduché údajové typy

- **Celé číslo (integer):** Uchováva celé čísla bez desatinných miest. Používa sa na počítanie, reprezentovanie ID, poradových čísel a iných celých hodnôt. Zaraďuje sa medzi jednoduché typy, pretože zaberá fixnú pamäť a jeho operácie (sčítanie, odčítanie, ...) sú efektívne.
- **Desatinné číslo (float):** Uchováva čísla s desatinnou časťou. Používa sa na reprezentáciu hodnôt, ktoré nie sú celé, ako napríklad ceny, merania a vedecké výpočty. Zaraďuje sa medzi jednoduché typy, aj keď oproti celým číslam zaberá viac pamäte a operácie s ním sú náročnejšie na výpočtový výkon.
- **Logická hodnota (boolean):** Môže mať iba dve hodnoty: TRUE alebo FALSE. Používa sa na reprezentáciu logických podmienok a testovanie pravdivosti či nepravdivosti výrazov. Zaraďuje sa medzi jednoduché typy pre svoju kompaktnosť a efektívnosť pri spracovaní logických operácií.
- **Znak (char):** Uchováva jeden znak, ako napríklad písmeno, číslicu alebo symbol. Používa sa na prácu s textom, ukladanie reťazcov znakov a reprezentáciu vstupu a výstupu. Zaraďuje sa medzi jednoduché typy pre svoju kompaktnosť a efektívnosť pri spracovaní textových dát.
- **Reťazec (string):** Uchováva sekvenciu znakov, čím tvorí text. Používa sa na prácu s textom, ukladanie viet, paragrafov a iných textových informácií. Zaraďuje sa medzi jednoduché typy, aj keď na rozdiel od jednotlivých znakov dokáže ukladať aj dlhšie texty. Zaberajú v pamäti fixný

počet bajtov, Operácie s nimi sú výpočtovo nenáročné, Majú definovaný typ (napr. int, float, bool, char, string), ktorý určuje ich vlastnosti a operácie.

a2 Charakterizujte reťazce, predvedte vytvorenie reťazca, vstup údajov do reťazca a volanie základných metód pre prácu s reťazcami

Do reťazca vieme ukladať textové hodnoty. Tieto hodnoty tam vieme buď priamo nastaviť alebo ich zobrať z klávesnice, z inputu užívateľa. Deklarácia reťazca:

```
string retazec;
```

Vytvorenie reťazca:

```
retazec = „Text“;
```

a3 Charakterizujte klasické pole, popíšte deklaráciu a vytvorenie poľa aj z hľadiska pamäte

Ukladajú kolekciu hodnôt rovnakého typu. Majú fixnú veľkosť definovanú pri vytvorení. Umožňujú priamy prístup k prvkom cez indexy. Prístup k prvkom je rýchlejší ako pri Liste.

Pole deklarujeme:

```
int[] pole;
```

Pole vytvoríme:

```
pole = new int[];
```

a4 Charakterizujte triedu ako údajovú štruktúru

- Definuje šablónu pre objekty: Trieda slúži ako návod na vytvorenie objektov s podobnými vlastnosťami a metódami. Určuje vlastnosti (atribúty) a metódy (operácie), ktoré objekty daného typu budú mať.
- Dedičnosť: Triedy dokážu dediť vlastnosti a metódy z iných tried (rodičovských tried), čím sa podporuje opakované použitie kódu a hierarchia typov.
- Polymorfizmus: Objekty rôznych tried, ktoré dedia z rovnakej rodičovskej triedy, dokážu implementovať metódy s rovnakým názvom, ale s rôznym správaním. To umožňuje flexibilitu a dynamické spracovanie objektov.

Trieda je základom objektovo orientovaného programovania. V triede definujeme vlastnosti objektu a čo objekt bude robiť, keď ho vytvoríme. Napríklad trieda Auto môže mať vlastnosti ako farba, značka, rok výroby atď. a metódy ako štartovanie, zastavenie, zrýchlenie a spomalenie. Je to ale iba v triede a žiadne také auto neexistuje, lebo sme ho ešte nevytvorili.

a5 Charakterizujte kontajnery a kolekcie ako úložiská údajov

Kontajner List v jazyku C# je štruktúra dát, ktorá umožňuje ukladať hodnoty rovnakého typu v jednom Liste. V liste vieme počas behu programu ukladať hodnotu a aj ich meniť. List môže meniť svoju veľkosť počas behu programu podľa potreby.

Deklarácia listu:

```
List<> priklad;
```

Vytvorenie listu:

```
priklad = new List<> ();
```

b1 Uved'te výhody používania kontajnerov a kolekcí oproti klasickým poliam

Nevýhody klasického poľa:

- Fixná veľkosť - nie je možné dynamicky meniť počet prvkov.
- Obmedzený typ prvkov - pole môže obsahovať len jeden typ dát.

List môže meniť svoju veľkosť počas behu programu podľa potreby narozdiel od pola.

b2 Predved'te použitie základných metód pri práci s kolekciami

Tu sú základné metódy pre prácu s kolekciami List:

- **Add:** nazovListu.Add(položka): Pridá položku na koniec listu.
- **Remove:** nazovListu.Remove(položka): Odstráni prvý výskyt zadaného prvku z listu.
- **RemoveAt:** nazovListu.RemoveAt(index): Odstráni položku na základe jej indexu v liste.
- **Clear:** nazovListu.Clear(): Odstráni všetky položky z listu.

Správa pamäte: V C++ je developer zodpovedný za manuálnu správu pamäte pre dynamicky alokované kontajnery. V C# sa o správu pamäte stará automatický garbage collector.

b3 Vytvorte program, ktorý naplní klasické N prvkové pole náhodnými číslami, pričom interval generovania nech je zadaný z klávesnice

Text

c1 Načrtnite topológiu s nutnosťou použitia Default Gateway

Text

c2 Popíšte základnú štruktúru podnikateľského plánu

Text

19. Programovanie – Bezparametrické a parametrické funkcie a metódy – Tomáš

a)

1. Uved'te, na čo slúžia funkcie a metódy a aký je medzi nimi rozdiel
2. Charakterizujte parametre, uved'te konkrétne príklady parametrov
3. Uved'te, ako sú volané funkcie a metódy bez návratovej hodnoty a ako s návratovou hodnotou
4. Vysvetlite, čo rozumieme pod preťažovaním funkcií a metód
5. Vysvetlite, čo rozumieme pod predefinovaním metód

b)

1. Vysvetlite, aký je rozdiel medzi statickými a nestatickými metódami
2. Vymenujte modifikátory prístupu v objektovom programovaní a uved'te, aký je medzi nimi rozdiel
3. Vytvorte program s dvomi triedami: jednu s metódou Main() a druhú s dvomi metódami – statickou na výpočet aritmetického priemeru piatich celých čísel a nestatickou na zistenie počtu slov vo vete

c)

1. Zdôvodnite potrebu protokolu ARP
 2. Vysvetlite úlohy a postavenie NBS, ktorá je súčasťou eurosystému
-

a1 Uved'te, na čo slúžia funkcie a metódy a aký je medzi nimi rozdiel

Funkcie a metódy sú extrémne dôležitou časťou programovania a dovoľujú nám zjednodušiť hľadanie chýb a písanie programu.

Pri písaní väčších programov nastanú mnohé problémy, a to ako sa v takom množstve kódy vyznať, kde nájsť chybu keď vznikne alebo jedna z nepríjemných vecí – napísanie tých istých 30 riadkov kódu stále dokola a dokola.

Funkcie:

- Je blok kódu, ktorý vykonáva určitú úlohu
- Môže byť volaná z akéhokoľvek miesta v programe.
- Funkcia môže mať parametre, ktoré sú hodnoty, ktoré sa predávajú do funkcie, aby ovplyvnili jej správanie.

Metódy:

- Je to špeciálny typ funkcie ktorý sa používa v objektoch alebo v triedach
- Metóda môže pristupovať len k len k dátam ktoré sú v tej danej triede
- Rozdiel medzi metódou a funkciou je, že metóda je volaná na konkrétnom objekte a funkcia môže byť volaná samostatne.

Funkcia slúži na vykonávanie rôznych úloh. Funkcie sa používajú na to, keď chcem viac krát spraviť určitú úlohu, tak to jednoducho zapíšem do funkcie. Funkcie si môžem vytvárať vlastné alebo používať už funkcie, ktoré existujú.

Ak nejaká funkcia ničomu nepatrí, je to funkcia. Funkciu si viem zavolať v maine a buď je funkcia bezparametrická alebo parametrická. Keď je s parametrami, tak jej dám do zátvoriek parametre.

Ak nejaká funkcia patrí objektu alebo triede, vtedy sú to metódy. Metódy patria triede.

a2 Charakterizujte parametre, uved'te konkrétne príklady parametrov

Parametrom je premenná zapísaná v hlavičke funkcie, ktorá zastupuje hodnotu predanú argumentom. Býva platný len v rozsahu funkcie. Funkcie môžu mať ľubovoľný počet argumentov a parametrov. Niektoré programovacie jazyky môžu vyžadovať, aby počet zadaných argumentov a parametrov pre funkciu bol rovnaký.

V niektorých programovacích jazykoch je potrebné v hlavičke funkcie definovať aj dátový typ parametra. V takomto prípade musia byť dátový typ argumentu a parametra rovnaké.

Druhy parametrov:

- **Povinný parameter:** tento parameter musí byť vždy zadany pri volaní funkcie alebo metódy v správnom poradí

- **Parameter s predvolenou hodnotou:** tento parameter má vo funkcii alebo metóde určenú nejakú hodnotu

Parameter je nejaká hodnota, ktorú môže mať funkcia. Napríklad existuje funkcia, ktorá vráti číslo.

```
function vratCislo(int cislo)
```

```
{ return cislo; }
```

A keď si ju volám, tak dám `vratCislo(parameter)` a do parametra intovú hodnotu.

Prípadne aj objekty môžu mať svoje vlastné parametre.

a3 Uvedzte, ako sú volané funkcie a metódy bez návratovej hodnoty a ako s návratovou hodnotou

Funkcia vracia buď skutočnú hodnotu, alebo prázdnu hodnotu typu `void`, teda nič. Funkcia, ktorá má napríklad sčítať dve celočíselné hodnoty, vracia hodnotu súčtu zadaných čísel a mala by byť deklarovaná s celočíselnou návratovou hodnotou.

Funkcia, ktorá má len vytlačiť správu na výstup, nemá čo vrátiť a mala by byť deklarovaná tak, že vráti hodnotu typu `void`.

Funkcie bez návratovej hodnoty sú `void`. Tieto funkcie na výstupe nič nevracajú. Potom existujú funkcie s návratovou hodnotou a tie už nám vrátia nejaký výsledok. Napríklad `int` alebo `string`. Napríklad funkcia `void` cez ktorú nastavíme hodnotu z nejakej premennej, napr. `void SetCislo(int cislo)`. Príklad návratovej hodnoty je `int GetCislo() { return cislo; }`

a4 Vysvetlite, čo rozumieme pod preťažovaním funkcií a metód

Je to viacero funkcií (metód) s tým istým menom, rozlišujú sa na základe počtu parametrov a typu parametra. Najčastejšie sa preťažovanie funkcií a metód používa v objektovo orientovanom programovaní. Preťažovanie sa používa pre zníženie počtu názvov metód, ktoré si je potrebné pamätať.

Preťaženie je možné vykonať nasledujúcimi spôsobmi:

- ak je definovaných viacero metód s rovnakým názvom, ale rôznymi parametrami.
- ak je definovaných viacero metód s rovnakým názvom a počtom parametrov, ale v odlišnom poradí.
- ak je definovaných viacero metód s rovnakým názvom, ktoré majú rovnaký počet a názvy parametrov, ale parametre sú rôzneho dátového typu.

Funkcie sa preťažujú vtedy, keď vytvoríme funkciu s rovnakým názvom, ale tej ďalšej funkcií s rovnakým názvom dáme iné parametre. O preťažovaní metód hovoríme vtedy, keď z rodičovskej triedy prevezmeme metódu do detskej triedy a pridáme tam ešte ďalšie parametre. Napríklad keď

máme triedu Byt, tak v nej môžeme mať metódu na GetCena() a v detskej triede je GetCena(), ale robí niečo iné.

a5 Vysvetlite, čo rozumieme pod predefinovaním metód

Zdedené metódy je možné v podtriede **predefinovať**:

Tzn. nad rôznymi objektami je možné zavolať tú istú metódu s tými istými parametrami, ale každý objekt sa bude chovať inak – ide o tzv. **polymorfizmus**.

Predefinovanie metód dosiahneme vďaka dedičnosti. Z rodičovskej triedy dedíme tak, že si detská trieda preberie z rodičovskej triedy hodnoty. A metódu predefinujeme tak, že v podradenej triede zmeníme jej parametre.

b1 Vysvetlite, aký je rozdiel medzi statickými a nestatickými metódami

Nestatické metódy:

- Majú prístup k dátam v objekte, v ktorom sú volané
- Sú volané na konkrétnej inštancii triedy a majú prístup k jej atribútom
- Môžu meniť stav objektu

Statické metódy:

- Nemajú prístup k dátam v konkrétnych inštanciách triedy
- Používajú sa na úlohy, ktoré sú spoločné pre všetky inštancie a nepotrebujú prístup k dátam objektu
- Nemajú prístup k žiadnym atribútom ani iným metódam objektu. Sú nezávislé od konkrétnej inštancie triedy

b2 Vymenujte modifikátory prístupu v objektovom programovaní a uveďte, aký je medzi nimi rozdiel

Private:

- Prístup iba z danej triedy
- Žiadna iná trieda nemôže pristupovať

Public:

- Prístup majú hoci ktoré triedy

Protected:

- Prístup majú iba potomkovia triedy (subclass, triedy ktoré túto triedu dedili)

Žiadny modifikátor:

- pokiaľ nie je uvedený žiadny modifikátor prístup majú len triedy v tom istom balíčku
- Je to významovo veľmi podobne ako keď je trieda private

- b3 Vytvorte program s dvomi triedami: jednu s metódou Main() a druhú s dvomi metódami – statickou na výpočet aritmetického priemeru piatich celých čísel a nestatickou na zistenie počtu slov vo vete

```
using System;

class Utility
{
    public static double CalculateAverage(params int[] numbers)
    {
        double sum = 0;
        foreach (int num in numbers)
        {
            sum += num;
        }
        return sum / numbers.Length;
    }
}

class MainClass
{
    public static void Main()
    {
        Console.WriteLine("Zadajte vetu:");
        string sentence = Console.ReadLine();
        int wordCount = CountWords(sentence);
        Console.WriteLine("Počet slov vo vete: " + wordCount);

        // Volanie statickej metódy pre výpočet aritmetického priemeru
        double average = Utility.CalculateAverage(1, 2, 3, 4, 5);
        Console.WriteLine("Aritmetický priemer čísel: " + average);
    }

    public static int CountWords(string sentence)
    {
        string[] words = sentence.Split(new char[] { ' ', '.', ',', '!', '?' },
        return words.Length;
    }
}
```

c1 Zdôvodnite potrebu protokolu ARP

Dynamické priradovanie IP adries:

- V sieťach je bežné používanie dynamických IP adries, kde sa IP adresy priradujú zariadeniam dynamicky pomocou protokolov ako DHCP

-ARP protokol umožňuje získanie MAC adries pre nové priradené IP adresy

Vyžaduje sa pri odosielaní dátových rámcov:

- Pri odosielaní dátových rámcov je potrebné poznať MAC adresu cieľového zariadenia

c2 Vysvetlite úlohy a postavenie NBS, ktorá je súčasťou eurosystému

-Určuje menovú politiku

-Vydáva bankovky a mince

-Riadi, koordinuje a zabezpečuje peňažný obeh, platobný styk a zúčtovávanie dát platobného styku v rozsahu ustanovenom týmto zákonom a osobitným zákonom a stará sa o ich plynulosť a hospodárnosť

20. Programovanie – Súbory – Zápis a čítanie - Adam

a)

1. Popíšte, na akom princípe funguje vstupno–výstupný súborový systém
2. Charakterizujte metódy a objekty používané pri súborových výstupoch a vstupoch
3. Charakterizujte jednotlivé typy súborov
4. Popíšte spôsoby zápisu údajov do súboru a spôsoby načítania údajov zo súboru
5. Uveďte, akými spôsobmi a metódami možno zistiť informácie o súboroch

b)

1. Popíšte serializáciu a deserializáciu objektov
2. Vytvorte program, ktorý do súboru zapíše náhodné čísla z intervalu zadaného z klávesnice
3. Dopĺňte program o výpočet aritmetického priemeru čísel, ktoré boli zapísané do súboru

c)

1. Vysvetlite štruktúru IPv6
2. Vysvetlite aké sú základné ekonomické voľby, ktorým čelí každá spoločnosť

a1 Popíšte, na akom princípe funguje vstupno–výstupný súborový systém

Vstupno–výstupný (I/O) súborový systém umožňuje prácu so súbormi v uložisku a funguje na princípe Streamu:

Stream (preložené do SK ako tok alebo prúd) je abstraktná trieda, ktorá poskytuje základné operácie vstupu a výstupu.

Funguje na dvoch hlavných konceptoch a to **Vstupný tok** (Input) – čítanie dát a **Výstupný tok** (Output) – zapisovanie dát.

Tieto operácie zahŕňajú čítanie, zapisovanie, presúvanie a zatváranie prúdu dát.

```
FileStream fstream = new FileStream("subor1.txt", FileMode.Create, FileAccess.Write);
```

a2 Charakterizujte metódy a objekty používané pri súborových výstupoch a vstupoch

Triedy **StreamReader** a **StreamWriter** sú určené na prácu s textovými súbormi zatiaľ čo triedy **BinaryReader** a **BinaryWriter** sú používané na binárne súbory.

```
using (StreamWriter sWriter = new StreamWriter(fstream))
```

Metódy v triede **File** – statické:

1. `.Create(string cesta);` – vytvorí nový súbor
2. `.Delete(string cesta);` – odstráni súbor
3. `.Exists(string cesta);` – zistí či súbor existuje (T/F)

4. `.Copy(string cestaZdroj, string cestaCiel);` – Kopíruje súbor
5. `.Move(string cestaZdroj, string cestaCiel);` – Presúva súbor

a3 Charakterizujte jednotlivé typy súborov

Typy súborov môžeme deliť podľa napríklad – štruktúry dát alebo formátu:

1. **Textové súbory** - Textové súbory sú základným typom súborov, ktoré obsahujú čitateľný text. Tieto súbory môžu obsahovať rôzne typy informácií, vrátane reprezentácie znakov, formátovaný text alebo dát vo forme ASCII alebo Unicode (reprezentácia znakov v počítačových systémoch).
2. **Binárne súbory** - Binárne súbory obsahujú dáta vo formáte, ktorý nie je priamo čitateľný pre človeka. Obvykle sa chápu ako sekvencia bajtov, čo znamená, že bity sú zoskupované po ôsmich. Obsahujú teda binárne dáta, ktoré môžu reprezentovať obrázky, zvuky, videá, spustiteľné programy, databázové súbory a ďalšie typy dát vo svojej neupravenej podobe.
3. **Obrázkové súbory** - Obrázkové súbory obsahujú grafické. Medzi najbežnejšie formáty obrázkových súborov patria JPEG, PNG, GIF.
4. **Zvukové súbory** - Zvukové súbory obsahujú zvukové dáta, ktoré môžu byť prehrávané pomocou zvukových prehrávačov alebo vložené do multimediálnych aplikácií. Medzi najbežnejšie formáty zvukových súborov patria MP3, WAV.
5. **Video súbory** - Video súbory obsahujú videodáta, ktoré môžu byť prehrávané pomocou videoprehrávačov alebo vložené do multimediálnych aplikácií. Medzi najbežnejšie formáty videových súborov patria AVI, MP4, MOV.

a4 Popíšte spôsoby zápisu údajov do súboru a spôsoby načítania údajov zo súboru

Zapisovať do súboru môžeme napr. pomocou tried **BinaryWriter** alebo **StreamWriter**. Najskôr si vytvoríme objekt triedy **FileStream**, do ktorého zadáme cestu k súboru, **FileMode** – Create(Vytvoriť) alebo Open(Otvoriť), a prístup **FileAccess** – Write, Read alebo ReadWrite.

Ďalej pomocou triedy **BinaryWriter** môžeme zapísať konkrétne dáta v binárnej forme. Alternatívne je možné použiť **StreamWriter** na zápis reťazcov.

Pri čítaní súborov si vytvoríme objekt triedy **BinaryReader** alebo **StreamReader**, pomocou ktorého môžeme čítať dáta zo súboru. Pri binárnych súboroch je nutné vedieť dátovú štruktúru súboru, ktorý sa snažíme prečítať. Ak štruktúru nepoznáme, prečítané dáta nemusia byť presné.

a5 Uved'te, akými spôsobmi a metódami možno zistiť informácie o súboroch

Informácie o súboroch je možné zistiť pomocou metód v triede **FileInfo**

Trieda **FileInfo**:

- a) **.Name** – vráti názov súboru
- b) **.Length** - vráti veľkosť súboru v bajtoch
- c) **.CreationTime** – vráti dátum vytvorenia súboru
- d) **.Exists** – vráti hodnotu T/F či súbor existuje
- e) **.Directory** – dostaneme adres

b1 Popíšte serializáciu a deserializáciu objektov

Serializácia a deserializácia sú procesy konverzie objektov v pamäti do formátu, ktorý je vhodný na uloženie alebo prenos, a následne spätná konverzia týchto dát späť na objekty.

Serializácia:

Pri serializácii sa objekty konvertujú do lineárneho formátu (prúd bitov idúcich za sebou), ktorý je zvyčajne vhodný na ukladanie do súborov alebo prenos cez sieť.

Použijeme triedu `BinaryFormatter`:

```
BinaryFormatter bFormatter = new BinaryFormatter();
```

Následne pomocou `FileStream` serializujeme objekt:

```
using (FileStream fStream = new FileStream("súbor.bin", FileMode.Create))
{
    bFormatter.Serialize(fStream, objekt);
}
```

Deserializácia:

Je opak serializácie, takže prúd bitov konvertuje späť na objekty s rovnakými hodnotami a štruktúrou, ktorú mali pred serializáciou. Typ a štruktúra objektu musí byť rovnaká ako pôvodný objekt.

```
using (FileStream fStream = new FileStream("súbor.bin", FileMode.Open))
{
    objekt = Convert.ToString(bFormatter.Deserialize(fStream));
}
```

b2 Vytvorte program, ktorý do súboru zapíše náhodné čísla z intervalu zadaného z klávesnice

```
0 references
static void Main(string[] args)
{
    a:
    Console.WriteLine("Zadajte dolnú hranicu intervalu: ");
    int dolnaHranica = Convert.ToInt32(Console.ReadLine());

    Console.WriteLine("Zadajte hornú hranicu intervalu: ");
    int hornaHranica = Convert.ToInt32(Console.ReadLine()) + 1;

    if(dolnaHranica >= hornaHranica)
    {
        Console.WriteLine("Zle zadané hranice");
        goto a;
    }

    Random rnd = new Random();
    using (FileStream fstream = new FileStream("cisla.bin", FileMode.Create, FileAccess.Write))
    using (BinaryWriter bWriter = new BinaryWriter(fstream))
    {
        for (int i = 0; i < 10; i++)
        {
            int nahodneCislo = rnd.Next(dolnaHranica, hornaHranica);
            bWriter.Write(nahodneCislo);
        }
    }
}
```

b3 Doplňte program o výpočet aritmetického priemeru čísel, ktoré boli zapísané do súboru

```
using (FileStream fStream = new FileStream("cisla.bin", FileMode.Open, FileAccess.Read))
using (BinaryReader bReader = new BinaryReader(fStream))
{
    int spolu = 0;
    int pocet = 0;
    while (bReader.BaseStream.Position < bReader.BaseStream.Length)
    {
        int number = bReader.ReadInt32();
        spolu += number;
        pocet++;
    }

    double priemer = spolu / pocet;
    Console.WriteLine("Aritmetický priemer čísel: " + priemer);
}
```

c1 Vysvetlite štruktúru IPv6

Internet Protocol verzie 6 je nasledovníkom súčasného Internet Protocolu verzie 4 a poskytuje rozšírenú adresáciu pre sieťové zariadenia a zlepšené funkcie siete.

1. Dĺžka adresy:

- IPv6 používa 128-bitové adresy, čo je výrazný nárast oproti IPv4, ktoré používajú iba 32-bitové adresy.

2. Adresný formát:

- Adresy IPv6 sú zvyčajne zapisované pomocou šestnástkovej číselnej sústavy: **2001:0db8:85a3:0000:0000:8a2e:0370:7334**.

3. Rozdelenie adresy:

- Adresy IPv6 sú rozdelené do troch hlavných častí:
 - **Prenosová adresa (prefix):** Určuje sieťovú časť adresy a je použitá na smerovanie paketov v rámci siete.
 - **Identifikátor podsiete (subnet ID):** Určuje konkrétnu podsieť, do ktorej patria zariadenia.
 - **Identifikátor zariadenia (interface ID):** Určuje konkrétne zariadenie v rámci danej podsiete.

4. Zjednodušené zápisy:

- IPv6 podporuje zjednodušené zápisy, ktoré umožňujú odstrániť zbytočné nuly z adresy a zredukovať zápis. Napríklad adresu **2001:0db8:0000:0000:0000:0000:1428:57ab** môžeme zapísať ako **2001:db8::1428:57ab**.

IPv6 je navrhnuté tak, aby umožnilo ďalší rast internetu a vyriešilo niektoré nedostatky IPv4, najmä nedostatok adresného priestoru. S jeho pomocou môžu sieťoví administrátori vytvárať veľké a efektívne siete a zabezpečiť, že internet bude schopný podporovať ďalší rast a rozvoj.

c2 Vysvetlite aké sú základné ekonomické voľby, ktorým čelí každá spoločnosť

Základné ekonomické voľby, ktorým čelí každá spoločnosť, sa týkajú alokácie obmedzených zdrojov s cieľom dosiahnuť určité ciele a maximalizovať zisk.

1. Rozhodovanie o výrobnom portfóliu:

- Spoločnosti sa musia rozhodnúť, ktoré výrobky alebo služby budú vyrábať a ponúkať na trhu. To môže zahŕňať rozhodnutia o diverzifikácii výroby, prípadne o sústredení na určitý typ produktu.

2. Rozhodovanie o výrobných procesoch:

- Spoločnosti musia zvoliť najefektívnejšie výrobné procesy, ktoré im umožnia minimalizovať náklady a maximalizovať produkciu.

3. Rozhodovanie o cene a cenovej stratégii:

- Spoločnosti sa musia rozhodnúť, akú cenu stanovia pre svoje výrobky alebo služby. Toto rozhodnutie ovplyvňuje ziskovosť a atraktivitu produktu pre zákazníkov.

4. Rozhodovanie o dodávateľoch a distribútoroch:

- Spoločnosti musia rozhodnúť, kde získajú suroviny a ďalšie potrebné materiály a ako ich distribuujú svoje výrobky alebo služby.

5. Rozhodovanie o investíciách a rozvoji:

- Spoločnosti musia rozhodnúť, kam investovať svoje zdroje s cieľom rozvíjať svoje podnikanie a zvyšovať svoju konkurencieschopnosť.

6. Rozhodovanie o ľudských zdrojoch:

- Spoločnosti musia riadiť svojich zamestnancov a rozhodnúť, ako najlepšie využiť ich schopnosti a zručnosti.

7. Rozhodovanie o financovaní:

Spoločnosti sa musia rozhodnúť, ako financovať svoju činnosť, či už prostredníctvom vlastného kapitálu, dlhov alebo iných finančných nástrojov.

21. Programovanie – Objektovo orientované programovanie - Marek

a)

1. Porovnajte objektovo orientované programovanie s procedurálnym programovaním
2. Charakterizujte základné pojmy OOP – trieda, členy tried, objekt
3. Vymenujte prístupové modifikátory a vysvetlite ich význam
4. Charakterizujte prístupové metódy
5. Uvedte, ako sa objekty konštruujú a deštruujú

b)

1. Popíšte, aký je rozdiel medzi nestatickými a statickými členmi tried

2. Charakterizujte dedičnosť, vysvetlite, akým spôsobom je volaný konštruktor bázevej triedy a čo znamená predefinovanie metódy
3. Vytvorte program, v ktorom z bázevej triedy odvodíte novú triedu

c)

1. Popíšte proces encapsulácie na PDU
2. Charakterizujte ekonomické a alternatívne ukazovatele celkovej úrovne bohatstva krajiny

a1 Porovnajzte objektovo orientované programovanie s procedurálnym programovaním

OOP funguje pomocou objektov a tried, využívame metódy, a konštruktory. V OOP je možné dedenie v rámci tried. **Polymorfizmus** – volanie jedného objektu s rôznymi parametrami

Procedurálne programovanie je založené na procedúrach kde je program rozdelený na súbor nezávislých procedúr, ktoré vykonávajú špecifické úlohy.

a2 Charakterizujte základné pojmy OOP – trieda, členy tried, objekt

Trieda – je samostatná časť kódu, ktorá slúži na vytvorenie inštancií a metód na ich úpravu

Objekt – je napr. inštancia triedy, má vlastné zadané hodnoty, vieme ich upravovať pomocou prístupových metód

Metóda – funguje podobne ako funkcia ale na jej zavolanie potrebujeme vytvorenú inštanciu triedy, s ktorou pracuje

Konštruktor – je v triede a slúži na vytvorenie inštancie triedy a zapísanie hodnôt

a3 Vymenujte prístupové modifikátory a vysvetlite ich význam

Public – Vieme ju volať z hocikákeho miesta v kóde

Private – Vieme volať iba z danej triedy

Protected – Vieme volať iba z danej triedy a z tried, ktoré dedia z triedy v ktorej je vytvorená

a4 Charakterizujte prístupové metódy

Prístupové metódy slúžia na prácu s objektami triedy. Napríklad si vytvoríme metódy get a set, tieto metódy budú slúžiť na výpis a zápis hodnôt ku konkrétnemu objektu.

a5 Uvedte, ako sa objekty konštruujú a deštruujú

Konštruktor a deštruktor – metódy na vytvorenie a zničenie inštancií tried. Konštruktor slúži na vytvorenie inštancie triedy a prípadné určenie základných hodnôt alebo hodnôt zadaných pri vytvorení.

Object dom1 = new dom(Okuliarova 6, 25, 4);

Deštruktor slúži na zničenie objektu a uvoľnenie pamäte

b1 Popíšte, aký je rozdiel medzi nestatickými a statickými členmi tried

Nestatické členy patria inštanciám triedy a voláme ich cez vytvorené objekty. Nestatické členy vedia používať statické.

Statické členy nepatria žiadnej inštancie triedy patria triede a voláme ju cez názov triedy. Nevedia používať nestatické členy.

b2 Charakterizujte dedičnosť, vysvetlite, akým spôsobom je volaný konštruktor bábovej triedy a čo znamená predefinovanie metódy

Dedičnosť funguje pri triedach, keď triedy vie zdediť z materskej triedy premenné.

Volanie konštruktora – Názov Triedy, „meno/nazov “ (parametre/premenné triedy)

b3 Vytvorte program, v ktorom z bábovej triedy odvodíte novú triedu

```
internal class Kniha
{
    protected int pocetStran;           //vytvorenie protected (aby ostatne triedy mohli dedit) premien
    protected double hodnotenieKritikov;
    protected string nazovKnihy;

    1 reference
    public Kniha () //bezparametricky konštruktor
    {
        pocetStran = 0;
        hodnotenieKritikov = 0;
        nazovKnihy = "Nezname meno";
    }

    2 references
    public Kniha(int pocetStran, double hodnotenieKritikov, string nazovKnihy) //3 parametricky konštruktor
    {
        this.pocetStran = pocetStran;
        this.hodnotenieKritikov = hodnotenieKritikov;
        this.nazovKnihy = nazovKnihy;
    }
}
```

```
internal class Hodnotenie : Kniha
{
    private double hodnotenieVerejnosti; // vytvorenie premennej

    //4 parametricky konštruktor
    1 reference
    public Hodnotenie(int pocetStran, double hodnotenie, string nazovKnihy, double hodnotenieVerejnosti) : base (pocetStran, hodnotenie, nazovKnihy)
    {
        this.hodnotenieVerejnosti = hodnotenieVerejnosti;
    }

    0 references
    public double GetHodnotenieVerejnosti()
    {
        return hodnotenieVerejnosti;
    }

    4 references
    public override double GetHodnotenie() // predefinovanie metody na získavanie hodnotenia knihy tak aby zpriemerovala hodnotenia kritikov a verejnosti
    {
        double hodnotenie = (hodnotenieKritikov + hodnotenieVerejnosti) / 2;
        return hodnotenie;
    }

    0 references
    public void SetMenoAutora(double hodnotenieVerejnosti)
    {
        this.hodnotenieVerejnosti = hodnotenieVerejnosti;
    }
}
```

c1 Popíšte proces encapsulácie na PDU

Sieťová vrstva zapuzdruje PDU z transportnej vrstvy do paketu. Proces encapsulácie pridáva informácie IP do hlavičky (IP adresa zdrojového a cieľového hostiteľa)

c2 Charakterizujte ekonomické a alternatívne ukazovatele celkovej úrovne bohatstva krajiny

HDP (hrubý domáci produkt) - Všetko čo sa v krajine vyprodukuje. Je to najčastejší ekonomický ukazovateľ používaný na meranie celkovej ekonomickej aktivity krajiny.

HDI (index ľudského rozvoja) - Tento ukazovateľ zohľadňuje nielen ekonomický výkon krajiny, ale aj ďalšie faktory ako očakávanú dĺžku života, úroveň vzdelania a životnú úroveň.

GINI koeficient - Tento ukazovateľ meria nerovnosť príjmov v krajine. Vyšší GINI koeficient zvyčajne naznačuje väčšiu nerovnosť v rozdelení príjmov medzi obyvateľmi.

Index udržateľného rozvoja - Tento ukazovateľ kombinuje aspekty ekonomickej výkonnosti, sociálneho rozvoja a environmentálnych faktorov.

Alternatívne ukazovatele - Okrem tradičných ekonomických ukazovateľov sa tiež stále viac používajú alternatívne ukazovatele ako napríklad Index šťastia, ktorý meria subjektívnu spokojnosť a pohodu obyvateľstva, alebo Index environmentálneho výkonu, ktorý hodnotí environmentálne politiky a ich dosah na životné prostredie.

22. Programovanie – Tvorba grafických rozhraní - Marek

a)

1. Porovnajte konzolové aplikácie s aplikáciami s grafickým rozhraním
2. Uveďte, čo rozumieme pod udalosťami riadenom programovaní
3. Charakterizujte komponenty z hľadiska OOP
4. Popíšte vývojové prostredie v režime tvorby aplikácie s grafickým rozhraním
5. Charakterizujte základné komponenty

b)

1. Vymenujte a popíšte základné vlastnosti komponentov
2. Popíšte obsluhu základných udalostí
3. Vytvorte aplikáciu s grafickým rozhraním, jednoduchú kalkulačku so základnými aritmetickými operáciami a do aplikácie doplňte funkciu na výpočet druhej odmocniny

c)

1. Popíšte PDU na transportnej vrstve
2. Charakterizujte banku a popíšte základné bankové operácie

a1 Porovnajte konzolové aplikácie s aplikáciami s grafickým rozhraním

Konzolové aplikácie využívajú buď terminál alebo konzolu na vstup a výstup a nevyužíva sa grafické rozhranie ale textové.

Aplikácie s grafickým rozhraním využívajú rôzne tlačidlá a komponenty a vstupy a výstupy su zobrazené graficky.

a2 Uved'te, čo rozumieme pod udalosťami riadenom programovaní

Výsledky dostávame na základe nejakej udalosti ktorá bola vyvolaná inou udalosťou na základe nejakých pravidiel / podmienok alebo rozhodovacích tabuliek.

a3 Charakterizujte komponenty z hľadiska OOP

Metódy - sú funkcie, ktoré sú súčasťou objektu a umožňujú manipuláciu s jeho stavom alebo vykonávanie operácií na ňom.

Zapuzdrenie - umožňuje schovanie / skrytie obsah premenných v objekte

Dedičnosť - umožňuje vytvárať nové triedy na základe existujúcich tried (nazývaných rodičovské triedy) a rozširovať ich funkčnosť a dediť ich premenné a funkcie.

Polymorfizmus - Polymorfizmus umožňuje spracovávať rôzne typy objektov rovnakým spôsobom.

a4 Popíšte vývojové prostredie v režime tvorby aplikácie s grafickým rozhraním

Editor kódu - IDE obsahuje editor kódu, ktorý umožňuje vývojárom písať, upravovať a organizovať zdrojový kód aplikácie.

Grafický editor pre dizajn GUI - Vývojové prostredie obsahuje nástroje na tvorbu grafického užívateľského rozhrania (GUI) prostredníctvom grafického editora.

Podpora pre udalosti a akcie - IDE poskytuje nástroje na definovanie interakcie medzi užívateľom a aplikáciou pomocou správy udalostí a akcií.

a5 Charakterizujte základné komponenty

Základné komponenty sú komponenty ktoré sú často využívané a môžeme ich vidieť v každej aplikácii.

Tlačidlá (Buttons) - spustia akciu po ich stlačení

Textové polia (Text Fields) - Textové polia umožňujú užívateľovi vstupovať textové údaje

Označovacie políčka (Checkbox) - Označovacie políčka umožňujú užívateľovi vybrať viacero možností naraz

Rozbaľovacie zoznamy (Dropdown Lists) - Rozbaľovacie zoznamy obsahujú možnosti, z ktorých užívateľ môže vybrať jednu

Zoznamy (Lists) - Zoznamy zobrazujú viacero položiek vo forme zoznamu

Panely (Panels) - Panel je oblasť, ktorá môže obsahovať iné komponenty GUI

b1 Vymenujte a popíšte základné vlastnosti komponentov

To isté čo a5 (myslím)

b2 Popíšte obsluhu základných udalosti

Nechápem tomuto // ani ja 😊

b3 Vytvorte aplikáciu s grafickým rozhraním, jednoduchú kalkulačku so základnými aritmetickými operáciami a do aplikácie doplňte funkciu na výpočet druhej odmocniny

Pracuje sa na tom

c1 Popíšte PDU na transportnej vrstve

PDU je na transportnej vrstve buď TCP segment (pre stránky – radšej pomalšie ale nesmú s stratiť data – email, sťahovanie) alebo UDP datagram (rýchle ale je akceptovateľná strata dát – live stream, volanie).

c2 Charakterizujte banku a popíšte základné bankové operácie

Text

23. Sieťové – Virtuálne siete LAN - Alex

a)

1. Definujte virtuálne siete
2. Zdôvodnite potrebu vytvorenia VLAN
3. Načrtnite dizajn VLANy
4. Popíšte access porty
5. Definujte trunk porty

b)

1. Vysvetlite konfiguráciu VLANy
2. Vysvetlite zapojenie Router-on-a- Stick, použite obrázok 1 obrazovej prílohy č. 23
3. Porovnaj router a multilayer switch, použite obrázok 2 obrazovej prílohy č. 23

c)

1. **Popíšte jednotlivé fázy podnikového transformačného procesu**
2. Vysvetlite pojmy konštruktor a prístupové metódy objektovo orientovanom programovaní

a1 Definujte virtuálne siete

Virtuálne siete (VLANs) sú logické skupiny zariadení v rámci fyzickej siete, ktoré komunikujú, akoby boli pripojené do samostatných sietí, aj keď zdieľajú spoločnú infraštruktúru.

a2 Zdôvodnite potrebu vytvorenia VLAN

Segmentácia siete: Umožňuje rozdelenie siete na logické skupiny, čo zvyšuje bezpečnosť, riadenie toku dát a výkon siete.

Bezpečnosť: Rozdelením siete na VLANy sa minimalizuje možnosť neoprávnenej komunikácie medzi rôznymi skupinami zariadení.

Optimalizácia výkonu: VLANy umožňujú izoláciu šírenia záťaže na rôzne segmenty siete, čím sa minimalizuje vplyv preťaženia na ostatné segmenty.

Správa siete: Umožňuje efektívnejšiu správu siete a priradenie zdrojov na základe potrieb skupín zariadení.

a3 Načrtnite dizajn VLANy

VLAN 10: Oddelenie pre oddelenie administratívnych zariadení.

VLAN 20: Oddelenie pre oddelenie oddelenia financií.

VLAN 30: Oddelenie pre oddelenie oddelenia marketingu

a4 Popíšte access porty

Access porty sú porty na switchoch, ktoré sú priradené len jednej určitej VLAN. Zariadenia pripojené k access portom komunikujú len s ostatnými zariadeniami v rovnakej VLAN.

a5 Definujte trunk porty

Trunk porty sú porty, ktoré prenášajú dáta z viacerých VLAN cez jeden fyzický kábel. Tieto porty sú typické medzi switchoch a medzi switchom a routerom. Slúžia na prenos dát medzi rôznymi VLANami.

b1 Vysvetlite konfiguráciu VLANy

Vytvorenie VLAN: Priradenie čísla a názvu VLANy.

Priradenie portov: Určenie, ktoré porty budú priradené k danej VLAN ako access porty.

Prípadne konfigurácia trunk portov: Ak je potrebné prepojiť viacero switchoch, je potrebné nastaviť trunk porty.

b2 Vysvetlite zapojenie Router-on-a-Stick, použite obrázok 1 obrazovej prílohy č. 23

Router-on-a-Stick je spôsob pripojenia switchoch k routeru cez jeden fyzický kábel (typicky trunk port). Router spracováva dátovú premávku medzi VLANami.

b3 Porovnajte router a multilayer switch, použite obrázok 2 obrazovej prílohy č. 23

Router: Pracuje na vrstve 3 modelu OSI, presmerováva dáta medzi rôznymi sieťovými segmentmi. Jeho funkcie zahŕňajú trasovanie a filtračnú funkciu.

Multilayer Switch: Kombinuje vlastnosti switchoch a routeru. Okrem prepojenia na vrstve 2 funguje aj ako router na vrstve 3 pre inter-VLAN komunikáciu. Je rýchlejší pre prenos dát medzi VLANami, pretože presmerovanie prebieha na hardvérovej úrovni.

c1 Popíšte jednotlivé fázy podnikového transformačného procesu

Analýza a plánovanie: Identifikácia potrieb a cieľov transformácie, plánovanie postupu.

Návrh a dizajn: Vytvorenie konkrétnych návrhov a stratégií pre transformáciu.

Implementácia: Vykonávanie zmien a implementácia nových procesov, technológií alebo štruktúr.

Monitorovanie a kontrola: Sledovanie a hodnotenie výsledkov transformácie, vykonávanie potrebných úprav.

Optimalizácia a udržiavanie: Nepretržité zlepšovanie a udržiavanie nových procesov a štruktúr.

c2 Vysvetlite pojmy konštruktor a prístupové metódy objektovo orientovanom programovaní

Konštruktor: Metóda v triede, ktorá je automaticky volaná pri vytvorení nového objektu tejto triedy. Používa sa na inicializáciu objektu.

Prístupové metódy: Slúžia na manipuláciu s dátami v objekte, umožňujú prístup a zmeny hodnôt atribútov objektu. Medzi ne patria metódy get (na získanie hodnôt) a set (na nastavenie hodnôt)

24. Sieťové – Protokol DHCP - Lukas

a)

1. Definujte pojem DHCP
2. Popíšte dôvody použitia DHCP
3. Vysvetlite postup komunikácie na získanie DHCP
4. Popíšte kroky pri konfigurácii DHCP, použite obrázok 1 obrazovej prílohy č. 24
5. Načrtnite príkazy na konfiguráciu DHCPv4

b)

1. Vysvetlite pojem DNS
2. Popíšte DHCPv6
3. Zdôvodnite potrebu použitia IP helper – address, použite obrázok 2 obrazovej prílohy č. 24

c)

1. Charakterizujte finančný trh, jeho funkcie a segmenty
2. Vymenujte a popíšte modifikátory prístupu v objektovo orientovanom programovaní, uveďte konkrétne príklady

a1 Definujte pojem DHCP

DHCP (Dynamic Host Configuration Protocol) je dynamický protokol pre konfiguráciu hostiteľov. Umožňuje počítačom a ďalším zariadeniam automaticky získať sieťové nastavenia, ako je IP adresa, gateway, maska siete a ďalšie, keď sa pripoja k sieti. DHCP poskytuje a priraduje informácie automaticky. DHCP funguje na princípe krátkeho pozicovania IP adresy, ktorá sa resetuje po ukončení práce.

a2 Popíšte dôvody použitia DHCP

Jednoduchá konfigurácia: nemusíme manuálne nastavovať sieťové parametre, ako napr. IP adresu, masku, gateway,...

Efektivnosť: DHCP pridelí zariadeniu IP adresu len dočasne, a tak sa nemenia až tak rýchlo adresy.

Mobilita: Tým, že DHCP automaticky priradí sieťové parametre, je výhodné hlavne pre mobilné zariadenia, ako napríklad notebook či smartfóny (zariadenia, ktoré sa často hybu medzi sieťami).

Spravovanie siete: Administrátor má väčšiu prehľadnosť nad pridelovaním adries, a aktivitou na sieti.

a3 Vysvetlite postup komunikácie na získanie DHCP

Žiadosť (DHCP Discover): Zariadenie, ktoré sa pripája k sieti a potrebuje IP adresu, pošle správu na DHCP server s požiadavkou na pridelenie IP adresy.

Ponuka (DHCP Offer): DHCP server reaguje na žiadosť odoslaním ponuky s dostupnými IP adresami a ďalšími sieťovými parametrami.

Výber (DHCP Request): Zariadenie si vyberá jednu z ponúkaných adries a posiela DHCP serveru žiadosť o túto konkrétnu adresu.

Potvrdenie (DHCP Acknowledge): DHCP server potvrdzuje výber zariadenia a pridáva mu pridelené IP adresy a sieťové parametre.

a4 Popíšte kroky pri konfigurácii DHCP, použite obrázok 1 obrazovej prílohy č. 24

1. Vylúčenie IP adries: Konfiguračné príkazy "ip dhcp excluded-address" sú použité na vylúčenie určitých IP adries z rozsahu, ktorý má byť priradený klientom DHCP.

2. Vytvorenie DHCP poolov: Konfiguračné príkazy "ip dhcp pool" sa používajú na definovanie jednotlivých DHCP poolov pre každú VLAN alebo sieť. V každom pooli sa definujú rozsahy IP adries, ktoré majú byť priradené klientom, ako aj ďalšie nastavenia, ako je výchozia gateway a možné ďalšie parametre.

V pooli "LAN-1" sú definované IP adresy pre sieť 192.168.2.0/24 s gateway 192.168.2.1.

V pooli "VLAN-10" sú definované IP adresy pre sieť 192.168.10.0/24 s gateway 31.0.2.1.

V pooli "VLAN-20" sú definované IP adresy pre sieť 192.168.20.0/24 s gateway 31.0.2.1.

V pooli "VLAN-30" sú definované IP adresy pre sieť 192.168.30.0/24 s gateway 31.0.2.1.

3. Uloženie a aplikácia konfigurácie: Po vytvorení DHCP poolov je potrebné uložiť a aplikovať konfiguráciu. Toto sa obvykle vykonáva príkazmi "write memory" alebo "copy running-config startup-config" na uloženie konfigurácie do pamäte.

Tieto kroky zabezpečia správne fungovanie DHCP a priradenie IP adries klientom v jednotlivých VLAN a sieťach.

a5 Načrtnite príkazy na konfiguráciu DHCPv4

1. Vytvorenie DHCP servera

```
Router(config)# interface <rozhranie>
```

```
Router(config-if)# ip address <IP_adresa> <maska>
```

```
Router(config-if)# ip dhcp pool <názov poolu>
```

```
Router(dhcp-config)# network <IP_siete> <maska>
```

```
Router(dhcp-config)# default-router <IP_východzej_gateway>
```

2. Priradenie DNS serverov

```
Router(dhcp-config)# dns-server <IP_DNS_serveru>
```

3. Špecifikovanie ďalších DHCP parametrov

```
Router(dhcp-config)# lease <čas_požičania>
```

4. Zapnutie DHCP servera

```
Router(config-if)# ip dhcp server
```

b1 Vysvetlite pojem DNS

DNS (Domain Name System) je systém, ktorý prekladá doménové mená (ako napríklad google.com) na numerické IP adresy, ktoré sú potrebné na identifikáciu zariadení a posielanie paketov na

internete. Je to v podstate adresár, ktorý umožňuje zariadeniam komunikovať a nájsť si cestu na internete.

b2 Popíšte DHCPv6

DHCPv6 je verzia DHCP pre použitie s IPv6. Podobne ako DHCP pre IPv4, poskytuje automatickú konfiguráciu sieťových nastavení pre zariadenia v sieti, ako sú IP adresy, gateway, maska,.... Umožňuje jednoduché a efektívne spravovanie IPv6 sietí.

b3 Zdôvodnite potrebu použitia IP helper – address, použite obrázok 2 obrazovej prílohy č. 24

1. Pre VLAN10 (192.168.10.0/24): Keďže sú v tejto VLAN dva počítače, ktoré potrebujú získať IP adresy od DHCP servera, ktorý je umiestnený na inom segmente siete (smerovač R_DHCP), je potrebné použiť IP helper-address. Bez neho by broadcastové DHCP požiadavky od počítačov v VLAN10 nepresiahli hranice VLAN a nedosiahli by DHCP server na smerovači R_DHCP.

2. Pre VLAN20 (192.168.20.0/24): Podobne ako v prípade VLAN10, aj tu je iba jeden počítač, ktorý potrebuje IP adresu od vzdialeného DHCP servera. Opäť je potrebný IP helper-address, aby broadcastové DHCP požiadavky z VLAN20 boli preposlané na DHCP server na smerovači R_DHCP.

S použitím **IP helper-address** môžu broadcastové DHCP požiadavky z oboch VLAN prejsť cez smerovač R_DHCP a dosiahnuť DHCP server, čo umožní priradenie IP adres počítačom v oboch VLAN. Bez IP helper-address by tieto požiadavky zostali obmedzené na jednotlivé VLAN a neuspeli by v získaní IP adres od vzdialeného DHCP servera.

c1 Charakterizujte finančný trh, jeho funkcie a segmenty

Finančný trh je miesto, kde sa stretávajú tí, čo majú peniaze, s tými, čo ich potrebujú. Jeho funkcie zahŕňajú alokáciu kapitálu, znižovanie rizika, poskytovanie informácií a podporu hospodárskeho rastu. Jeho segmenty zahŕňajú trhy dlhopisov, akcií, forex, komodít, realít, futures a opcií.

c2 Vymenujte a popíšte modifikátory prístupu v objektovo orientovanom programovaní, uveďte konkrétne príklady

Public: Tento modifikátor umožňuje prístup k atribútom alebo metódam z akéhokoľvek miesta v programe.

Príklad:

```
public string Meno;  
0 references  
public void metoda(string meno) { }
```

Private: Tento modifikátor obmedzuje prístup k atribútom alebo metódam len na vnútro triedy.

Príklad:

```
private double cislo;  
0 references  
private void metoda(string meno) { }
```

Protected: Prístup je obmedzený na triedy v rovnakej triednej hierarchii, ako aj na triedy v rovnakom balíčku. Atribúty alebo metódy s týmto modifikátorom sú prístupné aj v dedených triedach.

Príklad:

```
protected string meno;  
0 references  
protected void pozdrav() { }
```

25. ROB, SOA – Robotika a softvérové aplikácie - Dano

a)

1. Uveďte, čo je to Arduino, popíšte jeho základné časti
2. Popíšte využitie servomotora v praxi
3. Porovnajte pojmy robot a robotický manipulátor
4. Vysvetlite pojem kinematická štruktúra, použite prílohu č. 25
5. Vymenujte a popíšte typy podvozkov mobilných robotov

b)

1. Odvodte polomer otáčania diferenciálneho podvozku
2. Porovnajte optický inkrementálny a absolútny snímač polohy
3. Vytvorte program, v ktorom bude blikať LED s periódou 0.5s

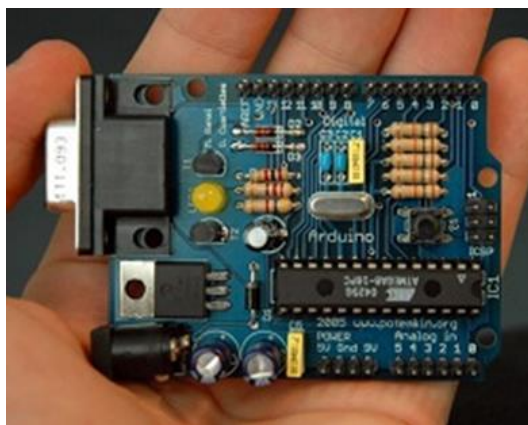
c)

1. Vysvetlite znaky, funkcie a právne formy podniku na Slovensku
2. Charakterizujte najpoužívanějšíe komponenty v aplikáciách s grafickým rozhraním

a1 Uveďte, čo je to Arduino, popíšte jeho základné časti

Arduino je názov malého jedno-doskového počítača založeného na mikrokontroléri ATmega od firmy Atmel. Svojim návrhom sa snaží podporiť výučbu informatiky v školách a oboznámiť študentov s tým, ako sú pomocou počítačov riadené rôzne zariadenia (napr. mikrovlnná rúra, automatická práčka a iné stroje).

Nejedná sa teda o počítač v zmysle stolného počítača alebo smartfónu. Nemožno preto k nemu ľahko priamo pripojiť monitor ani klávesnicu či myš, ale je pripravený na pripojenie LED diód, displeja z tekutých kryštálov, servomotorov, senzorov, osvetlenia, termistora, potenciometra



Arduino

Dosky Arduino obsahujú 8-bitové mikrokontroléry z rodiny AVR od firmy Atmel a množstvo ďalších podporných obvodov.

Základne časti Arduina Due:

Arduino Due obsahuje 32-bitový ARM procesor Atmel SAM3X8E. Každá doska má väčšinu I/O pinov prístupných cez štandardizované päťice, do ktorých sa jednoducho pripájajú ďalšie obvody, ktorým sa vo svete Arduino hovorí SHIELD. Na doskách býva niekoľko diód, resetovacie tlačidlo, konektory pre ICSP programovanie, napájací konektor, oscilátor a obvod sprostredkujúci komunikáciu po USB. Základná verzia Arduina (Arduino Uno), poskytuje celkom 14 I/O digitálnych pinov a 6 pinov analógových. Šesť digitálnych pinov je tiež možné použiť na softvérovo riadený PWM výstup.

Hlavný mikrokontrolér, ktorý je užívateľsky programovateľný, už má bootloader a nastavenie potrebných fuses bajtov. Vďaka tomu sa používateľ nemusí starať o detaily a svoje programy píše v jazyku podobnom C / C ++.

a2 Popíšte využitie servomotora v praxi

Servomotor sa v praxi využíva vo viacerých oblastiach ako napríklad:

V robotike sú servomotory kľúčovým prvkom, kde zabezpečujú plynulý a presný pohyb robotov. Ich schopnosť rýchlo a presne reagovať na signály zabezpečuje, že roboti môžu vykonávať rôzne úlohy s vysokou presnosťou a efektivitou. Táto presnosť je zásadná nielen pri priemyselnom využití robotov, ale aj v medicínskych aplikáciách, ako sú chirurgické zákroky, kde aj najmenšia chyba môže mať vážne následky.

Chirurgické roboty, ako napríklad da Vinci Surgical System, využívajú servomotory na ovládanie pohybu chirurgických nástrojov. Tieto systémy umožňujú lekárom vykonávať precízne a minimálne invazívne chirurgické zákroky s vysokou presnosťou a kontrolou.

V priemysle sa servomotory používajú na pohyb manipulátorov, ktoré manipulujú s ťažkými bremenami alebo vykonávajú presné montážne úlohy. Tieto manipulátory sú často súčasťou výrobných liniek v automobilovom, elektronickom a iných priemyselných odvetviach.

Automobily tiež používajú servomotory. V moderných automobiloch sa na riadenie rýchlosti používajú servomotory. Pri zošliapnutí plynového pedálu vysielajú elektrické signály do počítača automobilu. Počítač potom spracuje tieto informácie a odošle signál do serva pripojeného k škrtiacej klapke na úpravu otáčok motora.

a3 Porovnajete pojmy robot a robotický manipulátor

Rozdiel medzi manipulátorom a robotom je v tom, že manipulátor je v súlade s pevným alebo vopred nastaveným programom na napodobňovanie pohybu umelej ruky. Prostredníctvom rôznych trajektórií dokáže realizovať prepravu a manipuláciu s materiálmi v priemyselnej oblasti a zároveň dokáže obsluhovať nástroje na splnenie určených úloh. Na základe fixného programu manipulátora môže robot pomocou vlastných výkonových a inteligentných programov plniť učebné úlohy a inteligentné operácie, ktoré nie sú v rámci konkrétneho programu.

Manipulátor: Je to pevný alebo mobilný stroj, ktorého konštrukcia sa zvyčajne skladá zo série vzájomne prepojených alebo relatívne posuvných častí. Pracovný režim sa dosahuje hlavne lineárnym pohybom pozdĺž osí X, Y a Z na dosiahnutie cieľovej polohy. Uchopenie alebo premiestňovanie objektov môže realizovať automatické ovládanie, opakovateľné programovanie a viac stupňov voľnosti (DOF).

Manipulátor môže byť automatický alebo ovládaný človekom. Ide o elektromechanický systém, ktorý simuluje činnosť ľudských rúk a paží. Nemá žiadnu autonómnú schopnosť. Má jednoduché akcie a pevné prevádzkové postupy. Opakované operácie. Ovládacie zariadenie s nezmeneným polohovacím bodom. Manipulátor je široko používané mechanické zariadenie v oblasti robotiky, ktoré je široko používané v priemysle, zdravotníctve a dokonca aj vo vojenskom a iných oblastiach. Manipulátor je rozdelený na štyri osi, päť osí, šesť osí a viac osí, 3D/2D roboty, nezávislé manipulátory, hydraulické manipulátory atď.

Roboty: Často hovoríme o priemyselných robotoch. Priemyselné roboty sú automatizované zariadenia. Je veľa podôb a manipulátori sú len jednou z nich. Dá sa naprogramovať, ale zvyčajne je zmontovaný s koncovým efektorom (manipulátorom), aby vytvoril kompletný stroj. Priemyselné roboty sa pri vykonávaní práce spoliehajú na svoju vlastnú silu a riadiace schopnosti. Dokáže akceptovať ľudský príkaz, môže tiež bežať podľa vopred dohodnutých postupov a môže konať aj podľa princípov a smerníc formulovaných technológiou umelej inteligencie.

Priemyselné roboty majú vo všeobecnosti nezávislé ovládače, systémy pohonu a prevádzkové rozhrania, ktoré možno manuálne a automaticky ovládať a programovať. Ide teda o kompletne zariadenie, ktoré môže bežať samostatne a pomocou vlastného riadiaceho systému dokáže dosiahnuť požadované funkcie. Zariadenie má nezávislý mechanický mechanizmus a riadiaci systém, ktorý môže byť autonómny, komplexný v pohybe, väčšiu voľnosť práce, variabilné prevádzkové postupy a automatický operačný stroj, ktorý možno ľubovoľne polohovať.

a4 Vysvetlite pojem kinematická štruktúra, použite prílohu č. 25

Kinematika je štúdium vzťahu medzi súradnicami kĺbov robota a jeho priestorovým usporiadaním, je základnou klasickou témou v robotike. Kinematika môže poskytnúť veľmi presné výpočty v mnohých problémoch, ako je umiestnenie chápadla na miesto v priestore, návrh mechanizmu, ktorý dokáže presunúť nástroj z bodu A do bodu B alebo predpovedanie či by pohyb robota nenarazil na prekážky. Kinematika sa zaoberá iba okamžitými hodnotami súradníc robota a ignoruje ich pohyb pod vplyvom síl a krútiacich momentov. Problém kinematiky môže byť pre určité roboty dosť triviálny, ako sú mobilné roboty, ktoré sú v podstate tuhými telesami, ale vyžaduje si dôkladnú štúdiu pre iné roboty s mnohými kĺbmi, ako sú humanoidné roboty a paralelné mechanizmy.

Kinematická štruktúra robota je opísaným súborom spojov, ktoré sa pre väčšinu účelov považujú za tuhé telesá a spojov, ktoré ich spájajú a obmedzujú ich relatívny pohyb, napríklad rotačný pohyb.

Pri robotoch sa v praxi najviac rozšírili štyri **základné typy kinematických štruktúr**:

- | | |
|---|------------|
| 1. Tri translačné kinematické dvojice | TTT |
| 2 Jedna rotačná a dve translačné dvojice | RTT |
| 3. Dve rotačné a jedna translačná dvojica | RRT |
| 4. Tri rotačné dvojice | RRR |

a5 Vymenujte a popíšte typy podvozkov mobilných robotov

Existujú **tri hlavné typy podvozkov robotov**, menovite kolesový podvozok, pásový podvozok a podvozok s nohami:

Kolesový podvozok: Ako už názov napovedá, robot s kolesovým podvozkom používa na navigáciu kolesá. Kolesové podvozky sa dodávajú v mnohých rôznych tvaroch a veľkostiach a sú veľmi obľúbené vďaka svojej všestrannosti. Kolesové podvozky na robotoch sa časom stali pokročilejšími. Zatiaľ čo niektoré podvozky s kolesami sú navrhnuté len na pohyb v priamom smere, sú aj roboty s podvozkom na kolieskach, ktoré dokážu preliezť určité prekážky – vďaka čomu sú obzvlášť skvelé na vonkajšie použitie.

Hlavnou nevýhodou robotov s kolesovým podvozkom je, že môžu mať problémy s navigáciou v určitých typoch terénu. Pre niečo s kolesami je ťažké dostať sa na strmé alebo skalnaté povrchy, takže to dáva zmysel. Kolesové podvozky sú tiež o niečo menej stabilné ako iné typy podvozkov.

Pásový podvozok: Pásové podvozky zaznamenali nárast popularity vo svete robotiky kvôli spôsobu, akým sú schopné používať nepretržité trendy a dráhy na pohyb vo svojom prostredí. V porovnaní s kolesovým podvozkom majú roboty s pásovým podvozkom vynikajúcu mobilitu, čo znamená, že sa môžu ľahko pohybovať v množstve rôznych terénov. Majú tiež skvelú trakciu a sú super odolné, takže ak máte ťažkú prácu, ktorú musíte urobiť, robot s pásovým podvozkom je vynikajúcim uchádzačom.

Aj keď sú pásové podvozky lepšie, pokiaľ ide o navigáciu, v porovnaní s inými typmi podvozkov sú trochu pomalé. Sú tiež zložitejšie, čo znamená, že budú väčšinou drahšie. Pásový podvozok môže byť kvôli svojej komplexnej konštrukcii ťažší a menej energeticky účinný ako kolesový podvozok.

Podvozok s nohami: Ak ste niekedy videli robota, ktorý má štyri nohy ako pes, pravdepodobne má podvozok s nohami. Podvozok s nohami má špičkovú mobilitu a môže dokonca vyliezť po schodoch - čo je pre robota dosť pôsobivé. Podvozok s nohami sa tiež môže pohybovať vo viacerých rôznych smeroch a je schopný vykonávať zložité pohyby, ako je chôdza a skákanie.

Ako ste možno uhádli, roboty s podvozkom s nohami majú dosť komplikovaný dizajn. To môže často sťažiť a predražiť ich údržbu. Majú tiež tendenciu byť dosť pomalé, takže nie sú najlepšie, ak potrebujete urobiť prácu rýchlo s pomocou robota. Roboty s nohami majú tiež problém udržať rovnováhu na podlahách, ktoré sú hladké alebo klzké.



Typy podvozkov

b1 Odvodte polomer otáčania diferenciálneho podvozku

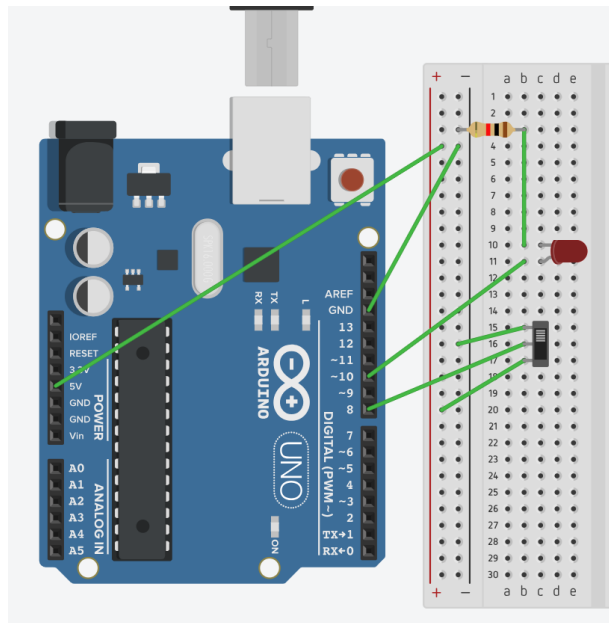
Polomer otáčania diferenciálneho podvozku môžeme odvodiť na základe jeho kinematickej štruktúry a vlastností pohybu. Diferenciálny podvozok je zložený z dvoch kolies, ktoré sa môžu otáčať nezávisle od seba. Polomer otáčania je vzdialenosť medzi stredovým bodom otáčania a stredovým bodom podvozku.

b2 Porovnajte optický inkrementálny a absolútny snímač polohy

Absolútny snímač vám môže povedať presnú polohu hriadeľa pri jeho otáčaní v akomkoľvek danom čase (a koľko otáčok sa uskutočnilo na viacotáčkovom absolútnom snímači). Inkrementálny snímač môže hlásiť iba zmenu polohy. Nemusí to znieť ako veľký rozdiel, ale ak má systém výpadok energie, je to deň a noc.

Po obnovení napájania môže absolútny snímač hlásiť presnú polohu bez akéhokoľvek pohybu. Inkrementálny snímač bude musieť nájsť index alebo domácu pozíciu, aby tak mohol urobiť – a ak index nemá, nebude vám vedieť povedať svoju polohu vôbec.

b3 Vytvorte program, v ktorom bude blikat LED s periódou 0.5s



```
#include <Servo.h>

Servo myServo;
int led = 10;

void setup()
{
  pinMode(led, OUTPUT);
}

void loop()
{
  digitalWrite(led, HIGH);
  delay(100);
  digitalWrite(led, LOW);
  delay(500);
}
```

c1 Vysvetlite znaky, funkcie a právne formy podniku na Slovensku

Podnikanie - sústavná činnosť vykonávaná podnikateľom samostatne, vo vlastnom mene, na vlastnú zodpovednosť s cieľom dosiahnuť zisk

Podnikateľom môže byť FO alebo PO, ktorá zakladá podnik prevádzkuje jeho činnosť.

Fyzická osoba je každý človek.

Právnická osoba - spoločenský útvar, ktorý vytvoril jednotlivec alebo jednotlivci, prípadne organizácie

Znaky podnikateľa:

- vykonáva činnosť sústavne a samostatne
- určuje predmet podnikania
- nesie podnikateľské riziko
- podniká na vlastný účet
- hospodári s majetkom
- jeho základným cieľom je dosahovať zisk

Právne formy podnikania:

1. Štátne podniky - vlastníkom majetku je štát

2. Živnosti - zvyčajne malé a stredné podniky, ktoré pôsobia vo sfére obchodu, služieb a remesiel

3. Obchodné spoločnosti - sú PO, ktoré sú založené za účelom spoločného podnikania alebo podnikania vo väčšom rozsahu

- osobné spol. - verejná OS, komanditná OS

- kapitálové spol. - spol. s r. o., akciová spol.

4. Družstvá - spoločenstvo neuzavretého počtu osôb založené za účelom podnikania alebo zabezpečovania hospodárskych, sociálnych alebo iných potrieb svojich členov

- stavebné družstvá, bytové družstvá, roľnícke družstvá, výrobné družstvá

5. Osobitné formy podnikania - nemajú právnu subjektivitu, nezapisujú sa do OR (obchodný register), nemôžu ako celok ručiť za záväzky voči tretím osobám

- Tiché spoločenstvo

6. Podnikanie zahraničných osôb - zahraničné osoby môžu podnikáť na území SR za rovnakých podmienok a v rovnakom rozsahu ako slovenské osoby

Funkcie podniku:

a) podnikateľská funkcia – základom tejto funkcie je sústavné a cieľavedomé využívanie podnikového kapitálu a tvorivého potenciálu na maximalizáciu zisku pri predchádzajúcom uspokojení potrieb zákazníkov,

b) ekonomická funkcia – zahrňuje finančnú, zásobovaciu a odbytovú funkciu. Poslaním týchto funkcií je hodnotovo vyjadriť hmotno-energetické procesy, vyčíslieť finančný výsledok, zabezpečiť podnik hmotnými a finančnými prostriedkami a zisťovať vplyv odberateľov na podnikovú organizáciu,

c) biologicko-ekologická funkcia – úlohou tejto funkcie je zabezpečovať tvorbu a ochranu životného prostredia, výrobu kvalitných potravín a ekologických výrobkov spotrebného priemyslu,

d) výrobná funkcia – prostredníctvom tejto funkcie sa zabezpečuje hlavná náplň činnosti podniku. Vo väčšine prípadov výrobná funkcia podmieňuje aj obsah a rozsah kooperačných a obchodných vzťahov,

e) organizačná funkcia – vyjadruje relatívne stálu činnosť smerujúcu k dosiahnutiu maximálneho stupňa organizačného usporiadania subjektu,

f) technická funkcia – patrí sem zabezpečenie výroby pracovnými silami a hmotným majetkom, a to z hľadiska kvality a kvalifikácie, zaradenie do organizačných jednotiek tak, aby sa dosiahli podnikovo-hospodárske ciele podniku,

g) sociálna funkcia – jej poslaním je zabezpečiť sociálnu efektívnosť výroby, zabezpečiť zamestnanosť pracovníkov a vytvárať podmienky na uplatnenie ich rozumových schopností, na zvýšenie ich odbornej a morálnej úrovne a na zvýšenie celkovej sociálnej úrovne podniku,

h) samosprávna funkcia – jej poslaním je prostredníctvom samosprávnych orgánov zabezpečiť autoreguláciu podniku.

c2 Charakterizujte najpoužívanejšie komponenty v aplikáciách s grafickým rozhraním
V **grafických používateľských rozhraniach (GUI)** sa niektoré prvky bežne používajú kvôli ich efektívnosti pri prenose informácií a interakcii s používateľmi. Tu sú niektoré z najčastejšie používaných prvkov:

Tlačidlá(Button): Tlačidlá sú základnými prvkami v GUI, s ktorými používatelia interagujú pri vykonávaní akcií. Môžu byť štylizované rôznymi spôsobmi, napríklad rôznymi farbami, tvarmi a

veľkosťami, aby zodpovedali celkovej estetike dizajnu. K tlačidlám sa často pridávajú textové štítky alebo ikony, ktoré označujú ich účel alebo činnosť. Tlačidlá zvyčajne reagujú na vstup používateľa, ako sú kliknutia alebo klepnutia, a spúšťajú súvisiacu akciu. Bežné prípady použitia tlačidiel zahŕňajú odosielanie formulárov, navigáciu medzi stránkami alebo spúšťanie špecifických funkcií v rámci aplikácie.

Textové polia(TextBox): Textové polia umožňujú používateľom zadávať textové alebo číselné údaje do GUI, ako sú používateľské mená, heslá alebo vyhľadávacie dopyty. Môžu byť jednoriadkové alebo viacriadkové, v závislosti od očakávaného vstupného formátu a dĺžky. Zástupný text sa často používa na poskytovanie rád alebo pokynov používateľom o očakávanom vstupe. Textové polia môžu mať overovacie pravidlá, ktoré zabezpečia, že zadané údaje budú spĺňať špecifické kritériá, ako sú povinné polia alebo správne formátovanie. Používatelia môžu pracovať s textovými poľami tak, že do nich píšú priamo pomocou klávesnice alebo vstupného zariadenia.

Štítky (Label): Štítky sú statické prvky používané na zobrazenie textu alebo titulkov v GUI, ktoré používateľom poskytujú kontext alebo pokyny. Často sú umiestnené vedľa iných prvkov GUI, aby opísali ich účel alebo poskytli dodatočné informácie. Štítky možno upraviť tak, aby zodpovedali celkovému dizajnu rozhrania, napríklad úpravou veľkosti písma, farby alebo zarovnania. Zvyčajne sú neinteraktívne a nereagujú na vstup používateľa. Bežné prípady použitia štítkov zahŕňajú štítky formulárov, popisy polí alebo nadpisy sekcií v GUI.

Začiarkavacie políčka(CheckBox): Začiarkavacie políčka umožňujú používateľom vybrať jednu alebo viac možností zo zoznamu možností ich zaškrtnutím alebo označením. Bežne sa používajú pre nastavenia, preferencie alebo otázky s možnosťou výberu z viacerých odpovedí, kde si používatelia môžu vybrať viacero možností. Každé začiarkavacie políčko predstavuje samostatnú voľbu a používatelia môžu svoj výber prepínať kliknutím alebo klepnutím na ne. Začiarkavacie políčka je možné zoskupiť a usporiadať súvisiace možnosti alebo ich môžete prezentovať jednotlivo na základe kontextu. Po vykonaní výberu môžu používatelia zvyčajne pokračovať so zvolenými možnosťami odoslaním formulára alebo potvrdením svojich volieb.

Prepínače(Radio Button): Prepínače umožňujú používateľom vybrať jednu možnosť zo zoznamu vzájomne sa vylučujúcich možností kliknutím alebo klepnutím na ne. Často sa používajú, keď používatelia potrebujú urobiť jeden výber z preddefinovanej sady možností. Na rozdiel od začiarkavacích políčok, kde je povolených viacero výberov, je možné vybrať naraz iba jednu možnosť prepínača. Prepínače sú zvyčajne usporiadané vertikálne alebo horizontálne, aby jasne prezentovali dostupné možnosti. Po výbere prepínača môžu používatelia zvyčajne pokračovať vo výbere odoslaním formulára alebo potvrdením svojho výberu.