

Výpočtový systém

Pod výpočtovým systémom rozumieme súbor hardwarových a softwarových prostriedkov, ktoré umožňujú spracovať vstupné dáta zadané užívateľom podľa jeho požiadaviek a vygenerovať výstupné dáta, ktoré užívateľovi slúžia ako podklad pre jeho rozhodovanie.

Modularita a hierarchická štruktúra systému

Z hľadiska štruktúry systému môžeme pozorovať, že každý takýto systém sa vyznačuje istou modularitou a hierarchickou štruktúrou.

Modularita

Každý súčasný výpočtový systém je riešený formou modulov, tzn. funkčných častí, ktoré plnia jednotlivé úlohy. Moduly vzájomne spolupracujú, dokážu si vzájomne odovzdávať dáta na ďalšie spracovanie a vzájomne komunikujú. Modularitu môžeme vysledovať na hardwarovej aj softwarovej úrovni.

Modularita na hardwarovej úrovni

Modularitou rozumieme, že systém pozostáva z jednotlivých modulov, ktoré vzájomným prepojením vytvárajú väčší funkčný celok. Funkcionalita tohto celku však predstavuje viac ako jednoduchý súčet funkcií jednotlivých modulov: spojenie modulov do celku poskytuje novú kvalitu.

Príklad:

za moduly môžeme považovať základnú dosku počítača, grafickú kartu, sieťovú kartu. Každý modul poskytuje určitú funkcionálnu, ale až ich prepojenie do funkčného celku – počítača – poskytuje novú kvalitu (možnosť realizovať výpočty, zobrazovať výsledky práce atď.)

Príklad:

samostatné počítače môžeme považovať za moduly, ktoré prepájame do vyššieho celku – počítačovej siete. Prepojenie počítačov do siete prináša celkom novú kvalitu – množstvo funkcií, ktoré samostatné počítače poskytovať nemôžu.

Príklad:

prepojenie počítačových sietí s telekomunikačnými sieťami a možnosť prístupu mobilných telefónov a smartfónov do Internetu prináša celkom novú funkcionálnu – novú kvalitu poskytovaných služieb.

Modularita umožňuje zo štandardizovaných komponentov postaviť zariadenie, ktoré je svojimi vlastnosťami prispôbené požiadavkám zákazníka, ďalšou výhodou je jednoduchá rozšíriteľnosť – časom je možné pridávať ďalšie komponenty, a zjednodušuje sa aj servis – oprava sa redukuje na diagnostiku, vyhľadanie chybného komponentu a jeho náhradu. Dôležitou vlastnosťou modularity je **interoperabilita** - schopnosť vzájomnej spolupráce, a **kompatibilita** – vzájomná spolupráca a zameniteľnosť modulov.

Príklady:

Interoperabilita – dáta uložená na disku pripojenom cez SCSI rozhranie je možné prostredníctvom počítača prekopírovať aj na ATA disk. Ak však počítač nie je vybavený SCSI rozhraním, treba najskôr vybaviť počítač potrebným rozhraním napr. formou karty.

Kompatibilita – pevné disky pripojené cez ATA rozhranie je možné pripojiť spoločne na jediný ATA kanál základnej dosky, sú vzájomne zameniteľné a formát dát pri prenose cez rozhranie je rovnaký.

Príklad:

Pevný disk bol pôvodne typickou súčasťou počítača, ale v súčasnosti ako komponent tvorí pamäťový prvok aj v DVD rekordéroch, alebo v digitálnych kamerách.

Podobnosti medzi počítačom a inými digitálnymi zariadeniami na úrovni použitých komponentov, ale aj princípov, sú zrejmé.

Modularita na softwarovej úrovni

Modularita sa uplatňuje aj na úrovni software: Najnižšiu úroveň tvoria elementárne príkazy programovacieho jazyka, z nich sa vytvárajú základné funkčné jednotky ako napr. ovladače HW zariadení, jadro operačného systému a pod., v rámci počítača potom všetky jednotky vzájomne spolupracujú a takto môžeme pokračovať aj po rozsiahle distribuované systémy, kedy na jedinej úlohe vzájomne spolupracujú celé rozsiahle výpočtové systémy zložené z tisícok počítačov.

Aj u software uľahčuje modulárne riešenie výstavbu celého systému, je možné ľahšie postupne dopĺňať jednotlivé komponenty systému, uľahčuje sa modernizácia systému – netreba modernizovať celý systém odrazu, ale modernizujú sa postupne jednotlivé moduly, a v neposlednom rade umožňuje modulárny systém spoluprácu viacerých programátorov, resp. tímov programátorov pri tvorbe väčšieho systému.

Modulárny systém umožňuje efektívne konštruovať rôzne zariadenia, ktoré využívajú vhodné komponenty, ale ich určenie je z hľadiska funkcie celkom odlišné. Príkladom môže byť napr. flash pamäťový modul, ktorý sa využíva na uloženie BIOS, firmware, obrázkov v digitálnom fotopapráte, hudby v MP3 prehrávači či adresy a telefónnych čísiel v mobilnom telefóne. Modulárny systém uľahčuje aj konštrukciu multifunkčných zariadení, ktoré v sebe združujú napr. funkcie vreckovej kalkulačky, mobilného telefónu, prehrávača hudby, diktafónu, záznamníku, kalendára a počítača.

Hierarchia systému

Okrem toho, že jednotlivé časti sú riešené formou modulov, môžeme vysledovať aj **rozvrstvenie** týchto **modulov** od najjednoduchších elementárnych súčiastok po zložité systémy s miliardami súčiastok **do určitých vrstiev**, pričom každá vrstva predstavuje z určitého pohľadu funkčný celok na istej úrovni.

Ako príklad si zoberme hardware počítača.

Jednotlivé elektronické prvky – súčiastky – vykonávajú elementárne úkony (napr. tranzistory pracujú ako zosilňovače alebo spínače), tým, že sú zapojené vo väčších celkoch - v obvodoch. Obvody vytvárajú jednotlivé komponenty (pamäťové jednotky, rozhrania a podobne) a komponenty sú kompletované do funkčných celkov (napr. počítač).

Z jednotlivých funkčných celkov je potom možné vytvárať väčšie systémy, pričom **ich výkon nemusí byť iba súčtom výkonov jednotlivých častí, ale celkový efekt sa môže násobiť**. Metóda konštrukcie zariadenia, kedy sa elementárne prvky skladajú do obvodoch, obvody do funkčných celkov – komponentov, komponenty do zariadení a zariadenia do systémov, nazývame **vrstvový model hierarchie**.

Hierarchický model podmieňuje jednak modularitu, ale aj opakované využitie komponentov či zariadení v rôznych funkciách.

Príklad:

Počítač je tvorený komponentami, ktoré sú vzájomne zameniteľné. Systém je možné doplniť o ďalšie komponenty- rozšírenie funkčnosti. Vybavenie PC WEB kamerou umožní používať PC ako zariadenie na záznam videa, alebo na videokomunikáciu.

Príklad:

Základná doska môže byť vybavená jediným CPU s jediným jadrom, ale dnes je bežné používať viacjadrové CPU a u výkonných počítačov viaceru spolupracujúcich CPU na jedinej základnej doske. Spojenie viacerých počítačov v rámci clusteru významne zvýši výpočtový výkon celku, a to nielen ako súčet výpočtových výkonov jednotlivých prvkov, ale celková efektívnosť systému sa násobí. Clustery je možné prostredníctvom vysokorýchlostnej siete prepojiť do ešte výkonnejších celkov.

Vrstvový model hierarchie je dôležitý aj vtedy, keď hodnotíme funkciu zariadenia na rôznych úrovniach. Iný pohľad bude mať elektronik – obvodár, ktorý bude merať obvodové veličiny a hodnotiť funkčnosť zariadenia na úrovni elektronických prvkov a obvodoch, iný pohľad bude mať technik – hardwarista, ktorý bude hodnotiť funkčnosť jednotlivých komponentov (HD,

grafická karta), iný pohľad bude mať programátor – vývojár alebo správca systému, ktorý hodnotí funkčnosť komponentov na úrovni funkčnosti výpočtového systému ako celku.

Pozn.:

Vrstvový model poznáme napríklad zo sieťových technológií – model OSI, model TCP/IP, hierarchia modelu IoT a ďalšie.

S každou vyššou vrstvou v hierarchickom modeli prichádza nová funkcionálna („užitočnosť“). Príklady.

Škálovateľnosť

Predstavuje schopnosť jednoduchými metódami rozširovať systém o nové časti alebo nové komponenty a tým zvyšovať jeho užitnú hodnotu.

Otvorený systém vs. uzatvorený systém

Otvorený systém – použité technológie sú verejne dokumentované, na rozširovaní systému sa môže podieľať ktorýkoľvek výrobca, ktorý dodržiava príslušné štandardy

Uzatvorený (proprietárny) systém je tvorený iba jediným výrobcom, (príp. konzorciom firiem) ktorý si parametre štandardov chráni ako firemné tajomstvo. Na vývoji a rozširovaní systému sa nepodíela žiadna iná firma okrem pôvodného výrobcu .

Vývoj výpočtových modelov

Prístupy ku výpočtovému systému

Z akých komponentov sa skladá výpočtový systém

- ☞ hardware
- ☞ software
- ☞ dáta
- ☞ systém komunikácie a prenosu dát
- ☞ užívateľ
- ☞ operátor

Hardware

Predstavuje **technické prostriedky** výpočtového systému.

Medzi architektúrou hardware a architektúrou výpočtového modelu existuje úzka väzba. Dôležitá je aj úloha počítačovej siete vo vzťahu ku výpočtovému systému. Napríklad celkom rozdielna úloha siete bude vo vzťahu ku osamotenému nezosieťovanému PC (sieť nie je potrebná) alebo ku mainframu, ktorý prostredníctvom siete poskytuje výpočtový výkon viacerým užívateľom (bez siete nie je práca možná)..

Software

Predstavuje **programové prostriedky** výpočtového systému. Je tvorený operačným systémom a aplikačnými programami. Z hľadiska výpočtového modelu je rozhodujúce, či je operačný systém koncipovaný ako

- ☞ jednoúlohový či viacúlohový
- ☞ jednougívatel'ský alebo viacugívatel'ský
- ☞ či podporuje alebo nepodporuje prácu po sieti

Tvorcom programu je programátor, dnes skôr tím programátorov.

Dáta

Dáta sú hlavným predmetom záujmu činnosti užívateľa. Dáta sú

- ☞ vstupné
- ☞ výstupné

Vstupné dáta predstavujú údaje, ktoré vstupujú v priebehu práce do výpočtového systému a sú spracovávané programom. Na základe programu a vstupných dát sú generované výstupné dáta.

Dáta môžu byť distribuované – rozptýlené v rôznych častiach výpočtového systému, alebo centralizované.

Príklad

Príkladom rozptýlených dát môže byť informačný systém, ktorý z rôznych zdrojov na Internete vyhľadáva relevantné údaje a spracúva ich do požadovanej podoby, alebo riadiaci systém ktorý sníma údaje z rôznych snímačov a na základe získaných veličín a vloženého programu riadi chod určitého zariadenia.

Príkladom centralizovaných dát môže byť databáza výrobkov v sklade alebo databáza žiakov školy.

Pri nevhodnej manipulácii s dátami v nesprávne koncipovanom výpočtovom modeli môže vzniknúť **dátový chaos** – viacero rôzne upravených kópií toho istého dátového súboru.

Dátovému chaosu sa dá predísť dôsledným dodržiavaním pravidiel **centralizovanej správy dát**.

Centralizovaná správa dát je nevyhnutná vtedy, ak viacero užívateľov pracuje nad spoločnými zdieľanými dátami.

Príklad

Príkladom centralizovanej správy dát môže byť databáza žiakov školy, databáza skladu prístupná súbežne viacerým používateľom, databáza podnikového informačného systému.

Nie vždy je však centralizovaný systém správy dát potrebný a vhodný – napríklad nikdy nie je možné centralizovať na jediný server všetky dátové zdroje, týkajúce sa určitej tematiky, ktoré sú rozptýlené po Internete. V takom prípade je nutné použiť **vhodný mechanizmus vyhľadávania dát z rozptýlených zdrojov**. V súčasnosti patrí medzi najsilnejšie nástroje v tomto smere Google.

System komunikácie a prenosu dát

Niektoré výpočtové modely, založené na centralizovanej práci jediného výpočtového systému, vôbec s existenciou systému komunikácie a prenosu dát nerátajú (dávkové spracovanie, osamotený nezosieťovaný PC)

Iné modely, ako je napríklad model klient – server, alebo grid computing, sú na využívaní prenosových prostriedkov priamo závislé.

Štandardne slúži na účely prenosu dát medzi počítačmi a ich vzájomnú komunikáciu počítačová sieť.

Užívateľ výpočtového systému

Je ním človek, prípadne tím pracovníkov, ktorý pre svoju prácu alebo svoje rozhodovanie potrebuje výsledky práce výpočtového systému.

Príklady:

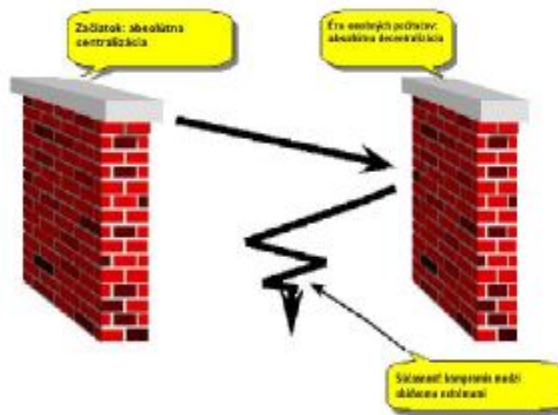
konštruktér lietadla – výsledky výpočtov profilu krídla,

študent – výsledky vyhľadávania údajov o výpočtových modeloch na Internete.

Za určitých okolností sa môže za užívateľa výpočtového systému považovať aj stroj alebo iný mechanický systém (automatizované systémy riadenia, NC stroje a pod.).

Operátor

Človek, ktorý **priamo manipuluje s ovládacími prvkami počítača** a bezprostredne pripravuje úlohu na spracovanie. V bežných moderných systémoch splyva pojem užívateľa s pojmom operátor – vďaka „user friendly“ užívateľskému prostrediu moderných výpočtových systémov stačí na ovládanie počítača prirodzená inteligencia a intuícia doplnená základmi IKT gramotnosti. V minulosti, v ére sálových počítačov, alebo aj **u moderných sálových počítačov typu mainframe či superpočítačov, býva fyzický prístup užívateľov k samotným výpočtovým prostriedkom obmedzený či celkom zabránený** a samotnú manipuláciu s počítačom vykonáva operátor, resp. tím operátorov.



Čo rozumieme pod pojmom výpočtový model

Je to **presný popis štruktúry systému vo vzťahu hardware – software – dáta – užívateľia**. Popisuje architektúru výpočtového systému ako celku.

Vo **vzťahu ku hardware** sa hodnotí koľko jednotiek systém vytvára, v akom režime spolupracujú, **ku software** – kde je uložený, kde sa spracováva, ako je koncipovaný, z koľkých častí sa skladá, ako je vyriešená spolupráca jednotlivých častí software

ku dátam – kde sú uložené, kde sa spracovávajú,

ku užívateľom – akú formu prístupu ku výpočtovým prostriedkom majú, ako je zabezpečené súkromie užívateľov, čo je zdieľané atď., ako môžu komunikovať s aplikáciami, aký majú prístup k dátam atď.

Výpočtový model

rieši:

- ☞ vzťah medzi výpočtovým systémom a užívateľom
- ☞ mieru centralizácie / decentralizácie výpočtového modelu
- ☞ rozdelenie úloh medzi jednotlivé moduly
- ☞ metódy uloženia dát a prístup výpočtového modelu ku dátam
- ☞ metódy prístupu užívateľov ku počítaču a komunikáciu užívateľov s počítačom

Vzťah výpočtový model – počítačová sieť.

Každý model má iný vzťah ku sieti a kladie na sieť iné nároky. Pri návrhu siete je nevyhnutné pochopiť, pre aký výpočtový model je sieť navrhovaná, čo model od siete bude požadovať a jej architektúru týmto podmienkam prispôbiť. Niektorý model nepočíta s existenciou siete, **väčšina modelov je však založená na dátovej komunikácii medzi viacerými výpočtovými uzlami**. Dôležité je nielen fyzické prepojenie hardwarových komponentov, ale aj vzájomný vzťah softwarových modulov.

Prehľad výpočtových modelov

Jedným z najdôležitejších parametrov výpočtového modelu je miera centralizácie HW a SW prostriedkov a dát. Vývoj výpočtových modelov prechádzal

- ☞ od absolútne centralizovaných modelov
- ☞ cez absolútne decentralizovaný model
- ☞ až ku súčasným modelom, ktoré sa vyznačujú istou mierou centralizácie, ale súčasne aj decentralizáciou výpočtových prostriedkov a softwarových modulov

Dávkové spracovanie

Historicky ide o najstarší model. Jediný, na svoju dobu výkonný počítač, je vlastnený a spravovaný inštitúciou, ktorá zabezpečuje jeho efektívne využívanie. Počítač pracuje lokálne, nepoužíva zdieľané zdroje, všetky komponenty výpočtového systému sú sústredené na jedinom mieste – v počítačovej sále.



6

Tento model je dobre použiteľný na **riešenie úloh náročných na výpočty** – vedeckotechnické výpočty, riešenie výskumných úloh, hromadné spracovanie veľkých objemov dát, spracovanie štatistických úloh so spracovaním veľkého počtu vzoriek a podobne.

Nie je vhodný na úlohy ktoré vyžadujú interaktivitu, úlohy s vysokou mierou on-line práce. **Užívateľ nemá prístup ku počítaču.** Zadanie úlohy postúpi programátorovi a v spolupráci s ním vypracuje tzv. dávku. Dávka obsahuje príslušný program a súbor vstupných dát, uložených na vhodnom médiu (vonkajšia výmenná pamäť).

Dávka je odovzdaná operátorovi počítača, ktorý zariadi vykonanie úlohy na počítači. Výsledok je vhodnou formou (spravidla v podobe tlačovej zostavy alebo súboru dát uložených na vhodnom médiu) odovzdaný zadavateľovi – užívateľovi.

Výhody:

- ☞ strojový čas si môžu prenajať aj inštitúcie, ktoré by si kúpu vlastného veľmi výkonného a drahého počítača nemohli dovoliť
- ☞ drahé hardwarové prostriedky sú využité s vysokou efektívnosťou
- ☞ počítač môže využívať viacerí používatelia, aj keď neexistuje HW ani SW podpora multitaskingu
- ☞ vysoká ochrana HW prostriedkov pred zneužitím a poškodením (prístup ku počítaču majú iba špeciálne vyškolení operátori)

Nevýhody:

- ☞ nulová interaktivita

- ☞ nemožnosť v priebehu výpočtu reagovať na priebežné výsledky a prípadne ukončiť výpočet, ak je jeho pokračovanie zbytočné – operátori nie sú kompetentní rozhodovať o problémoch súvisiacich s predmetom výpočtu), nemožnosť v prípade potreby nechať výpočet uskutočniť opakovane s inými vstupnými hodnotami
- ☞ nemožnosť odlaďovať program priamo na počítači (nedá sa programovať metódou „pokús – omyl“, programátori musia vytvárať precízne programy, ktoré budú fungovať napoprvý krát)
- ☞ dlhý čas obrátky (užívateľ čaká na výsledok dni aj týždne)

Súčasnosť dávkového spracovania:

Využíva sa u veľmi drahých a výkonných počítačov, ktoré nie je možné z ekonomických dôvodov prevádzkovať iba pre jediného užívateľa (superpočítače či niektoré špecializované mainframy)

Na rozdiel od čias počítačovej „prehistórie“, kedy dávka bola predstavovaná sadou diernych štítkov s pripravenou úlohou, je dnes dávka distribuovaná po sieti do výpočtového centra, a tam čaká na zaradenie do frontu pre spracovanie. Niektoré systémy ponechávajú na užívateľovi, aby si vybral stroj, na ktorom bude úloha spracovaná, niektoré systémy samy vyberajú podľa určitých kritérií (typ úlohy, jej náročnosť na systémové vybavenie, obsadenie počítačov inými úlohami) vhodný systém na spracovanie. Výsledok je uložený na sieťovom serveri a užívateľ je upovedomený napr. e-mailom o vykonaní úlohy a údajoch ktoré sprístupnia výsledky.

Model host – terminál

Tento model historicky nasleduje po dávkovom modeli.

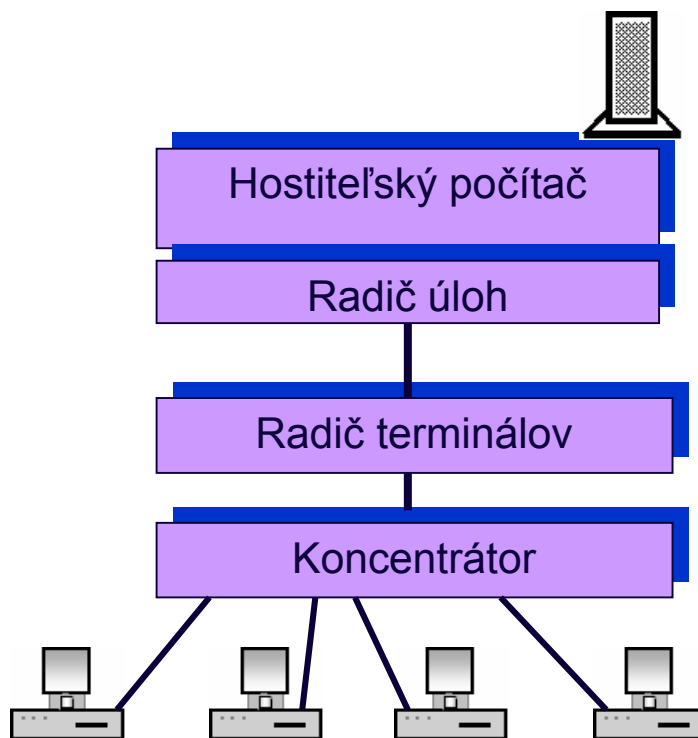
Výpočtový systém je predstavovaný jediným (na svoju dobu vždy veľmi výkonným) počítačom. Keďže je počítač „hostiteľom“ výpočtových prostriedkov – procesora, pamäte, vstupných a výstupných zariadení, ale aj operačného systému a aplikačného software, zaužíva sa pre hostiteľský počítač termín „host“. **Prístup ku počítaču je možný prostredníctvom terminálov.**

Terminál – klávesnica, monitor (v prehistorických časoch tlačiareň) a komunikačné zariadenie, ktoré zabezpečuje pripojenie terminálu ku počítaču.

S príchodom terminálov dochádza aj ku prelomovej udalosti v oblasti softwarového vybavenia – prichádzajú prvé operačné systémy, prostredie komunikácie užívateľov s počítačmi sa zjednocuje a zjednodušuje a obsluha počítača prestáva byť výsadou špecialistov – operátorov, ale je umožnená aj kvalifikovaným užívateľom.

Výhody:

- ☞ Prostredníctvom terminálu môže užívateľ priamo komunikovať s počítačom, zadávať mu úlohy a získavať výsledky. Z procesu sa vylučuje operátor.
- ☞ Interaktivita – užívateľ priamo komunikuje s počítačom. Možnosť opakovať výpočet bez čakania s inými dátami, možnosť rozhodovať o smere pokračovania výpočtu v určitých medzikrokoch, možnosť interaktívne ladiť program.



- ☞ Ak je počítač pripojený na banku dát, možnosť interaktívne pracovať s týmito dátami.

Postupne boli HW aj SW počítačov vybavené možnosťou **multitaskingu** – súbežného spracovania viacerých úloh. Počítač môže byť vybavený **viacerými terminálmi** a môže obsluhovať viacero užívateľov súčasne. Objavuje sa **zabezpečenie privátnych dát, logovanie do systému**, ostáva možnosť práce skupiny používateľov nad spoločnými zdieľanými dátami - **centralizovaná správa dát**.

Ďalší vývoj prináša vybavenie terminálov **zariadeniami na diaľkovú komunikáciu** – prichádzajú prvé modemy, ktoré umožňujú pripojenie terminálov prostredníctvom telefónnych liniek na vzdialenosti desiatok kilometrov. Objavujú sa **základy počítačových sietí**. Komunikačné rýchlosti sa pohybujú rádovo okolo 1000 byte za sekundu, na prenos znakov medzi terminálom a počítačom je rýchlosť dostatočná.

Výhody modelu host - terminál s diaľkovým prístupom

- ☞ dáta sú centralizované, nevzniká dátový chaos, dosahuje sa vysoká bezpečnosť uložených dát
- ☞ dáta sú uložené spolu so SW na jedinom počítači – nevznikajú problémy s potrebou prenosu dát pre výpočet
- ☞ malé nároky na prenos dát medzi terminálom a hosťom (iba požiadavky a výsledky)
- ☞ užívateľ sa nemusí starať o správu a bezpečnosť systému

Nevýhody modelu host – terminál

- ☞ užívateľ sa musí o strojový čas, prípadne o strojový výkon, deliť s inými užívateľmi. Počítač nemá „pre seba“ – je odkázaný na dostupnosť počítača, ktorý vlastní a spravuje niekto iný.
- ☞ terminál nedisponuje žiadnym výpočtovým výkonom, ak je počítač vypnutý alebo nedostupný, prípadne ak má počítač poruchu, nie je pre užívateľa k dispozícii žiadny náhradný výpočtový výkon.
- ☞ užívateľ nemá možnosť zasiahnúť do systémových nastavení počítača, nemá možnosť zvoliť si operačný systém, konfigurovať si prostredie
- ☞ užívateľ je odkázaný na danú HW konfiguráciu, nemá možnosť zasahovať do HW konfigurácie počítača
- ☞ nízka úroveň komfortu – prac. prostredie je znakové

Ide o prísne centralistický model – všetky prostriedky sú sústredené na **host**.

Súčasnosť a perspektíva: Model host – terminál je v súčasnosti väčšinou nahradený modelom klient – server, prípadne tenký klient – server.

Terminálové siete umožňujú prostredníctvom špeciálnych terminálových koncentrátorov sa pripojiť ku vzdialenému hosťovi.

Terminálové emulátory (špeciálne programy, ktoré umožňujú klasickému PC „sa tváriť“ ako terminál) umožňujú vzdialený prístup ku hosťom z klasických PC, ktoré vo vzťahu ku hosťovi vystupujú ako terminály.

Model samostatných osobných počítačov

Tento model vznikol v polovici sedemdesiatych rokov 20. storočia ako protiklad ku dovtedy používaným centralizovaným modelom. Jeho vznik umožnil masívny nástup relatívne lacných



(a porovnaní s vtedy bežnými sálovými počítačmi možno povedať, že veľmi lacných) **osobných počítačov** na trh.

Základom tohto modelu je množstvo úplne samostatných osobných počítačov, ktoré nie sú žiadnym spôsobom prepojené. Tento model je z pohľadu firmy príkladom **absolútnej decentralizácie**. Každý užívateľ disponoval **svojim vlastným počítačom**, so svojimi programami a so svojimi dátami. Nič nemusel zdieľať, o nič sa nemusel deliť s inými užívateľmi. Prenos dát medzi počítačmi bol možný iba pomocou výmenných pamäťových zariadení (diskety).

Príchod osobných počítačov predstavuje masívny nástup využitia počítačov pre bežné kancelárske práce (textové editory, tabuľkové kalkulátory, grafické editory, malé databázové systémy, DTP a pod.).

Pre užívateľa ako jedinca systém javí ako dokonale **centralizovaný** – všetko má na svojom pracovnom stole. Ovšem **z pohľadu firmy a správy dát, správy systémov a správy hardware ide o absolútnu decentralizáciu**.

Výhody:

Komfort – užívateľ mal svoj počítač vo svojej kancelárii na stole, mal ho k dispozícii stále, mohol si nainštalovať programy ktoré potreboval a sám si počítač obsluhoval.

Nevýhody:

- ☞ Užívateľ sa musel o počítač aj sám starať, musel si zvýšiť kvalifikáciu o operátora (musel sa naučiť pracovať v nie celkom jednoduchom prostredí príkazového riadku), a o správcu systému (musel sa naučiť inštalovať a udržiavať operačný systém aj aplikačné programy).
- ☞ Nastupuje častá nestabilita systému, havárie systému, straty dôležitých dát, úniky dôležitých dát do nepovoláných rúk – všetko v dôsledku nekompetentných zásahov nekvalifikovaných užívateľov.
- ☞ Často sa ten istý problém rieši opakovane na rôznych pracoviskách (znova a znova sa pracne objavuje to, čo bolo už mnohokrát objavené)
- ☞ Výmena dát medzi počítačmi bola pracná, nepohodlná a pri väčšom objeme dát aj zdĺhavá.
- ☞ Nebola možná centralizovaná správa dát, veľmi často dochádza ku dátovému chaosu.. Nebolo možné zdieľanie žiadnych prostriedkov, hardwarová zostava „na stole“ bola všetko, čo mal užívateľ k dispozícii.

Model osobných počítačov s použitím lokálnej siete peer – to – peer.

Princípom modelu je **pospájanie počítačov do siete bez servera** a **nastavenie zdieľania** niektorých prostriedkov na počítačoch. Počítače tak môžu pracovať v úlohe servera aj klienta. Tento model predstavuje vykročenie od prísne decentralizovaného modelu ku kompromisnému efektívnemu modelu a reaguje na hlavné nevýhody samostatných nezosieťovaných osobných počítačov.

Vyriešil komplikácie s častým **prenosom väčšieho objemu dát** a **so zdieľaním HW** prostriedkov a dát a **častočne** umožnil riešenie **centralizovanej správy dát** a **centralizovaného zálohovania**.

Nevýhody: Keďže HW ani SW týchto počítačov nie je konštruovaný na sieťovú prácu, bývajú problémy s dostatkom výkonu a s bezpečnosťou.

Významné bezpečnostné riziko predstavujú aj samotní používatelia, pretože ich oprávnenia v sieti typu P2P často významne presahujú ich kvalifikáciu.

Sieť typu P2P je použiteľná pre malý počet počítačov (cca do 10) a pre implementácie, ktoré nie sú náročné na požadovaný výkon a na bezpečnosť.

Výhodou oproti modelu klient – server je jednoduché, lacné riešenie, ktoré si nevyžaduje náročné investície do nákupu a správy servera.

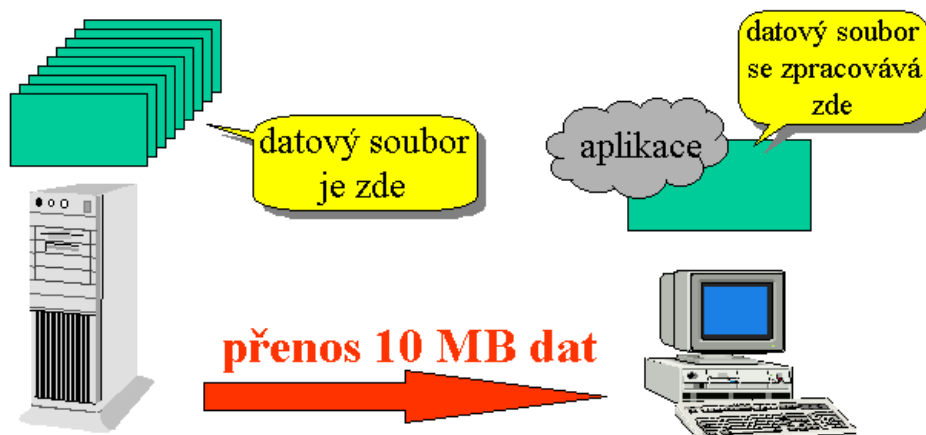
Model klient – file server.

Je to **najjednoduchší model siete s využitím servera**

Princípom je sieť so serverom, ktorý disponuje dostatočným diskovým priestorom na **uskladnenie dát užívateľov** a prípadne aj **časti software**. Na serveri sú umiestnené aj **hardwarové prostriedky, určené na zdieľanie**. Výkon servera a priepustnosť siete musia byť na takej úrovni, aby server dostatočne dokázal plniť úlohy klientov bez zdržiavania. V tomto modeli však **nedochádza ku vykonávaniu aplikačných programov na serveri** – dáta aj prípadné aplikačné programy sú po sieti stiahnuté na lokálnu stanicu, tam sú vykonané potrebné úlohy a výsledky sú uložené na serveri, alebo na lokálnej stanici – prípadne na oboch destináciách.

Příklad: práce s databází

- úkol: prohledat DB soubor velikosti **10 MB**,
zda se v něm nachází položka XY



38

Výhody:

- ☞ na serveri sú umiestnené spoločné dáta (centralizovaná správa)
- ☞ môžu tu byť aj aplikácie
- ☞ býva tu zálohovací systém
- ☞ býva tu diskový priestor na uloženie „súkromných“ dát
- ☞ môžu tu byť zdieľané HW prostriedky
- ☞ ochranu pred neoprávneným prístupom zabezpečuje systém prístupových práv systém pracuje spoľahlivo aj keď operačný systém na klientskej stanici nerozlišuje, či požadované prostriedky sa nachádzajú na lokálnej stanici či na serveri (mapovanie sieťových prostriedkov na písmená diskov lokálneho operačného systému)

Nevýhody:

Vysoké nároky na prenos veľkých objemov dát medzi klientami a serverom

Príklad:

Klient potrebuje vyhľadať jediný záznam v rozsiahlej databáze. databáza aj príslušný program sú uložené na serveri. Klient si musí stiahnuť celú rozsiahlu databázu aj príslušný program na svoju stanicu, vykonať požadovanú operáciu (napr. vyhľadanie jediného záznamu a jeho editáciu) a odoslať celú opravenú databázu späť na server. Potrebné dáta na operáciu predstavujú niekoľko bajtov, prenášajú sa desiatky aj stovky megabajtov. V čase editácie databázy je pre iných užívateľov na úpravy uzamknutá.

Príklad:

Program ani lokálny operačný systém nepodporujú prácu na sieti, boli navrhované pre nesieťovaný osobný počítač. Počítač bol dodatočne vybavený sieťovou kartou, príslušnými ovladačmi a klientom sieťového operačného systému. Požadované dáta sú uložené na serveri v priečinku „UCTOI“. Sieťový operačný systém mechanizmom mapovania priečinka na písmeno disku zariadi, že v operačnom systéme na lokálnej stanici sa objaví napr. disk „U:“, hoc fyzicky taký disk na počítači nejestvuje. Pod písmenom „U:“ sa ukrýva priečink „UCTOI“ na serveri, a operačný systém dokáže s dátami v tomto priečinku pracovať ako keby sa nachádzali fyzicky na lokálnom disku „U:“.

Model klient – aplikačný server.

Tento model vznikol z modelu klient – file server na základe jednoduchej úvahy: Server disponuje dostatočnou výpočtovou silou (alebo je vecou majiteľa siete, aby takýto server nainštaloval). Nie je potrebné prenášať na klientskú stanicu všetky dáta, a niekedy aj aplikačné programy, keď sa dá zariadiť, aby užívateľ iba vyslal ku serveru požiadavku, všetky potrebné výpočty, resp. iné operácie (vyhľadávanie) sa vykonajú na serveri, a na klientskú stanicu sa odošle iba výsledok operácie.



Významne sa šetrí potrebné množstvo prenášaných dát po sieti aj potrebný výpočtový výkon na strane klienta.

Princíp práce teda vyzerá nasledovne:

Klient vyšle požiadavku ku serveru, požiadavka predstavuje rádovo desiatky či stovky prenesených bajtov. Server vo svojich dátach, s využitím programu inštalovaného na serveri a s využitím vlastného výpočtového výkonu uskutoční požadované operácie a výsledok – rádovo desiatky až stovky bajtov – odošle po sieti na klientskú stanicu.

Príklad:

Moderné webové stránky využívajú viaceré technológie. HTML kód a Java scripty sú interpretované na strane klienta, PHP kód je vykonávaný na strane servera.

Tento princíp práce veľmi pripomína model terminál – host a v určitom zmysle predstavuje jeho použitie na novej úrovni.

Model tenký klient – aplikačný server.

Pre mnohé implementácie (verejné knižnice, počítačové učebne) sa ukázalo, že používanie plnohodnotných PC je pre tieto účely zbytočné – počítače sú zbytočne drahé. V týchto nasadeniach nie je potrebný vysoký výpočtový výkon, ale skôr je potrebný rýchly a pohodlný prístup ku dátovým zdrojom na serveri.

Model *host – terminál* bol „oprášeny“ a postavený na bezdiskových stanicích a software typu tenký klient. Ide o stanice bez akýchkoľvek vlastných diskových prostriedkov, vybavených iba základnou doskou s procesorom, operačnou pamäťou a výkonným sieťovým rozhraním. Súčasťou sieťového rozhrania je bootovací ROM modul, ktorý umožňuje naboťovať počítač priamo zo servera. Všetok software, vrátane operačného systému sa pri štarte počítača natiahne do jeho operačnej pamäte zo servera, potom však už stanica pracuje so serverom v podobnom režime ako terminál. Pre bezdiskové stanice sa preto vžil názov „inteligentný terminál“ (= terminál disponujúci vlastným procesorom a výpočtovým výkonom).

Aplikácie musia mať vždy časť inštalovanú na klientskej stanici a časť inštalovanú na strane servera. Vhodne riešené aplikácie potom dokážu potrebné výpočty rozdeliť na výpočty vykonávané na strane klienta a výpočty na strane servera.

Model distribuovaný systém

Distribuované systémy boli navrhnuté pre riešenie úloh, ktoré si vyžadujú **vysoký výpočtový výkon** a užívateľ nemá počítač s dostatočným výkonom k dispozícii. Iným podnetom je neefektívnosť sústredenia všetkých dát a aplikácií na jediný server.

Riešením je vytvorenie siete výkonných počítačov, riadených jediným operačným systémom a spolupracujúcich **na riešení jediného výpočtu**, koncipovať informačné systémy ako **moduly**, ktoré môžu fungovať samostatne, ale v prípade potreby aj vzájomne komunikovať. Celý systém je umiestnený na **viacerých počítačoch**, ktoré sú prostredníctvom výkonnej siete vzájomne prepojené a **voči užívateľovi dokážu vystupovať ako komplexný funkčný systém**.

Príklad

informačného systému riešeného ako distribuovaný systém

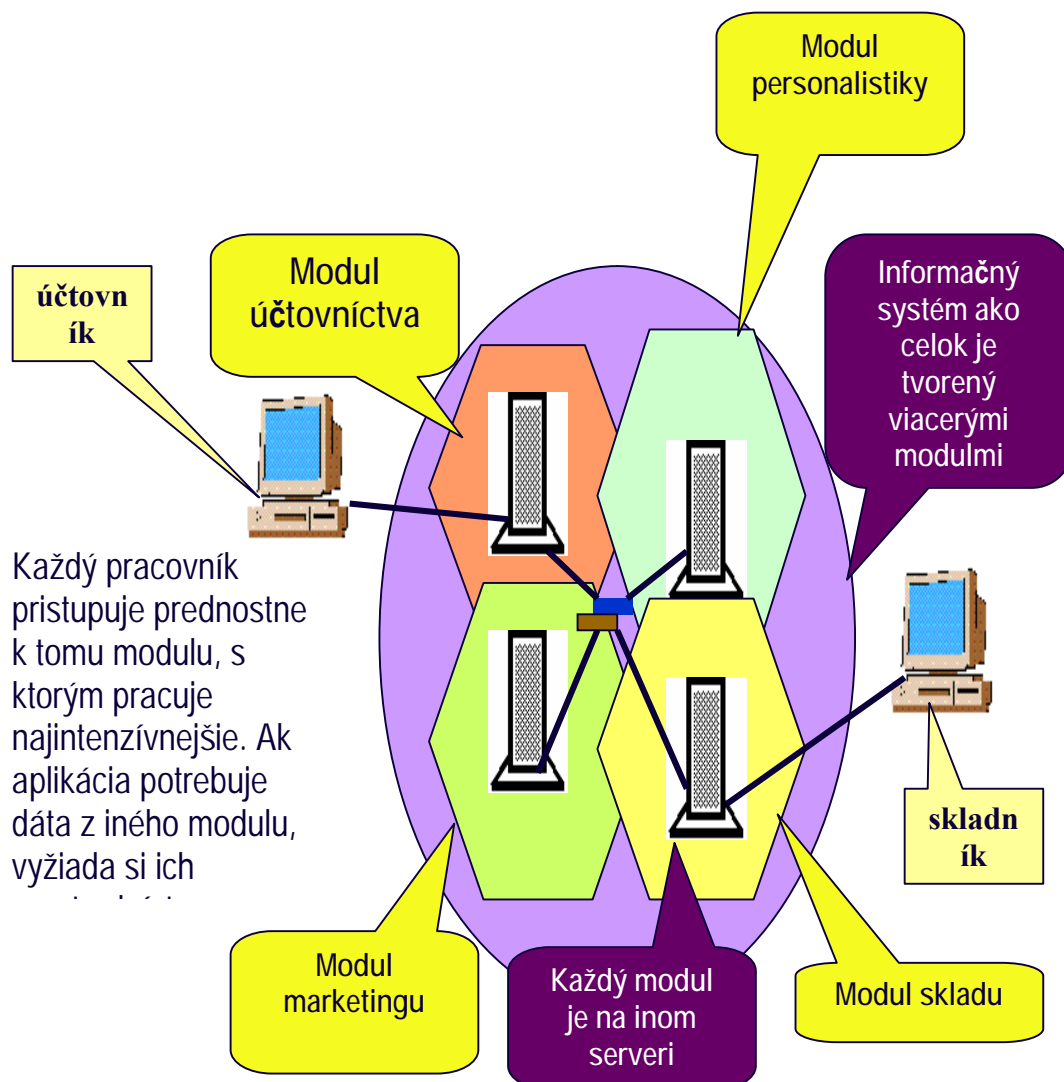
- ☞ Informačný systém ako celok je tvorený jednotlivými modulmi, navonok sa však javí ako jediný systém
- ☞ Každý modul je umiestnený na inom stroji (serveri)
- ☞ Stroje sú zosieťované a dokážu spolupracovať
- ☞ Zamestnanec sa hlási ku systému (nie k modulu), ale systém ho – podľa zaradenia – prihlási ku „jeho“ modulu
- ☞ Ak niektorý modul potrebuje pre vyriešenie zadanej úlohy dáta z iného modulu, zariadi ich dodanie informačný systém prostredníctvom siete

Distribuované systémy ako výkonné výpočtové systémy:

- ☞ ide o voľnejší model multiprocesingu, kedy sú jednotlivé počítače viazané prostredníctvom siete za účelom zvýšenia výpočtového výkonu a zvýšenia spoľahlivosti systému
- ☞ špeciálnym prípadom sú clusterové systémy, kde je väzba medzi počítačmi HW aj SW užšia, ako u distribuovaného systému, úlohy sú efektívnejšie rozdeľované a výkon clusterového systému sa blíži výkonu superpočítača (podľa mohutnosti HW)

Zhrnutie:

- ☞ distribuovaný systém je zdokonalením modelu klient – aplikačný server
- ☞ zachováva všetky výhody centralizovaného modelu
- ☞ je možné k nemu pristupovať z bežného PC, je možné ho orientovať na systém tenkých klientov



rozdelením systému na moduly a pridelením modulov ku samostatným strojom je možné rovnomerne rozdeliť záťaž, zefektívniť výkon a minimalizovať nároky na prenosy v sieti

Cloud computing

Predstavuje **d'alšiu generáciu vo vývoji výpočtových modelov**. Je kombináciou viacerých už spomínaných modelov – terminálového prístupu, modelu file server – klient, aplikačný server – klient aj distribuovaného systému.

Jadrom sú dátové centrá, vybavené veľmi výkonnými počítačmi a rozsiahlymi dátovými skladmi. Umiestnenie serverov v topológii siete nie je pre užívateľa zrejmé – centrum sa nachádza „kdesi v cloude“.

Zákazníkom sa poskytuje služba ako celok – provider poskytuje HW, príslušný SW, priestor na uloženie dát, zabezpečuje zálohovanie dát, stará sa o pripojenie a bezpečnosť.

Cloud computing predstavuje v súčasnosti smer, ktorým sa vývoj výpočtových modelov ubera, ovšem prináša so sebou aj špecifické riziká (zrútenie celej infraštruktúry pri strate konektivity, nebezpečenstvo kompromitácie veľmi citlivých dát, uložených v cloude).

Internet vecí (Internet of Things)

Cloud computing sa stal základným prvkom architektúry, nazývanej **Internet of Things**. Táto architektúra rozoznáva tri vrstvy:

Fog – techniku, ktorá zahŕňa senzory a akčné členy, kontroléry a ďalšiu výpočtovú techniku, ktorá priamo komunikuje so senzormi a koncovými zariadeniami siete a realizuje okamžité rozhodnutia.

Infrastructure – vrstva zabezpečujúca komunikáciu zariadení na úrovni fog s cloudom.

Cloud, ktorý slúži na zhromažďovanie, triedenie, ukladanie a analýzu dát.

Na základe výsledkov analýz sú potom z cloudu zariadením vo vrstve fog zasielané pokyny a príkazy na príslušné akcie. Akcie môžu mať charakter okamžitých reakcií, alebo dlhodobých strategických prístupov.

Agentové systémy

Charakteristika:

Na rozdiel od distribuovaného systému, ktorého dominantným „pracovným prostredím“ sú servery, je agentový systém určený prednostne na prácu na klientských počítačoch. S distribuovaným systémom má spoločné rozdelenie aplikácie na moduly, schopné samostatnej práce, ale v prípade potreby sú moduly schopné vzájomne spolupracovať. Moduly sú navyše schopné v sieti sa vzájomne vyhľadávať.

Ako agentový systém je charakterizovaný taký výpočtový systém, ktorý spĺňa nasledujúce požiadavky:

- autonómnosť: agent koná bez akéhokoľvek priameho ľudského alebo iného zásahu a má kontrolu nad svojimi akciami a vnútorným stavom
- sociálna schopnosť: agenti interagujú s inými agentami (prípadne človekom) prostredníctvom nejakého komunikačného jazyka
- reaktivita: agenti vnímajú svoje okolie (to môže predstavovať fyzický svet, používateľa - prostredníctvom grafického rozhrania, kolektív agentov, Internet alebo kombináciu spomenutých) a v príhodnom čase reagujú na zmenu vo svojom okolí
- pro-aktivita: konanie agentov nie je jednoduchá odozva na stav okolia, ale môže byť aj cieľovo orientované s prevzatím iniciatívy

Technológie agentových systémov využívajú veľmi často sieťové červy a počítačové vírusy

Grid computing

Charakteristika:

- ☞ je variantom agentového systému
 - ☞ dokáže využívať jednotlivé klientské počítače na riešenie úloh vyžadujúcich vysoký výpočtový výkon
 - ☞ na prácu využíva nečinné, ale spustené PC, pripojené vysokorychlostným pripojením k Internetu
 - ☞ dispečer (riadiaci server) zasiela jednotlivým PC úlohy a sústreďuje a triedi získané výsledky
- Pre aký typ úloh je tento model vhodný:**
- ☞ úlohy vyžadujúce vysoký výpočtový výkon
 - ☞ zároveň musia byť úlohy rozdeliteľné na veľké množstvo vzájomne nezávislých čiastkových výpočtov

Zhrnutie

Súčasný model predstavujú z hľadiska systému jediný celok, z hľadiska fyzického umiestnenia SW aj dát ide o využívanie viacerých zosieťovaných a vzájomne spolupracujúcich výpočtových jednotiek. Väčšina výpočtových modelov je založená na kooperácii počítačov – pracujú v niektorom z režimov multiprocesingu, vyžadujú efektívne fungujúcu sieť a SW schopný využívať distribuované zdroje

Otázky na opakovanie

- Aké poznáte modely prístupov ku počítačom?
- Aké sú výhody a nevýhody jednotlivých modelov?
- Aké výhody poskytuje dávkový model prístupu k výpočtovému systému ?
- Využíva sa výpočtový model dávkového spracovania aj v súčasnosti?
- Aké ekonomické a technologické dôvody viedli ku dávkovému modelu ?
- Prečo sa v súčasnosti používa dávkový model iba výnimočne?
- Pre aké typy úloh je vhodné použitie modelu dávkového spracovania?
- Pre aké typy úloh nie je dávkový model výhodný?
- Kde má aj v súčasnosti význam používanie dávkového modelu?
- Kde je dávkový model nepoužiteľný?
- V čom je prínos modelu host – terminál proti v dávkovému modelu?
- Ako súvisí nástup modelu host – terminál s rozvojom sieťových technológií?
- Aké sú nevyhnutné technologické predpoklady na umožnenie terminálového prístupu?
- Existuje variant modelu host – terminál, ktorý poskytuje možnosť prístupu ku hosťovi zo vzdialenej kancelárie?
- Ako súvisí model host – terminál s technológiami dátových prenosov cez telekomunikačné linky?
- Ako sa líši terminálová sieť od počítačovej siete?
- Aký je rozdiel medzi lokálnym terminálovým prístupom a vzdialeným terminálovým prístupom?
- Čo je potrebné k tomu, aby bežný PC mohol slúžiť ako vzdialený terminál?
- V čom je model klient – server podobný modelu host – terminál?
- V čom sa líši model klient – server od modelu host – terminál?
- Aké výhody prináša výpočtový model samostatných (nezosieťovaných) PC?
- Aké technologické predpoklady viedli ku vzniku osobných počítačov?
- Aké boli ekonomické súvislosti prudkého nástupu osobných počítačov?
- Aké nevýhody prináša výpočtový model samostatných (nezosieťovaných) PC?
- Ako sa vo výpočtovom modeli samostatných PC bez siete rieši komunikácia medzi počítačmi?
- Ako sa vo výpočtovom modeli samostatných PC bez siete rieši tlač v prípade, že je k dispozícii iba jediná tlačiareň pre skupinu používateľov?
- Ako sa vo výpočtovom modeli samostatných PC bez siete rieši problematika dátového chaosu?
- Aké dôsledky na bezpečnosť prináša nástup samostatných osobných počítačov?
- Čo prináša užívateľovi zosieťovanie osobných počítačov do siete peer-to-peer?
- Aké nevýhody prináša sieť P2P?
- Aké boli dôvody zosieťovania výkonných strediskových počítačov?
- Vznik akého typu sietí podmienila potreba zosieťovania strediskových počítačov?
- Aké nevýhody má riešenie typu file server – pracovná stanica?
- Prečo sa budujú siete P2P namiesto efektívnejších sietí klient-server?
- Aké výhody prinášajú siete klient-server v porovnaní so sieťami P2P?
- Aký je vzťah modelu terminál – host ku modelu tenký klient – server?
- Ktorý z modelov je možné označiť za maximálne centralistický?
- Ktorý z modelov je možné označiť za maximálne decentralistický?
- Ktorý z modelov je možné označiť za rozumný kompromis medzi úplnou centralizáciou a totálnou decentralizáciou?

- Ako musí byť postavený SW pre sieť klient – server, aby umožnil minimalizáciu prenosu dát medzi klientom a serverom?
- Musí v rámci jediného informačného systému bežať klient aj server pod rovnakým operačným systémom?
- Nakreslite blokovú schému trojvrstvého usporiadania modelu klient – server so samostatným strojom pre aplikačné funkcie a samostatný stroj pre správu dát. Uveďte konkrétny príklad z praxe.
- Aké momenty determinujú vznik lokálnych sietí na báze PC?
- Čo rozumieme pod distribuovanými systémami?
- Uveďte príklad a charakteristiku informačného systému riešeného ako distribuovaný systém
- Na príklade systému webových stránok vysvetlite princíp distribuovaného systému
- Vysvetlite hlavné princípy fungovania agentových systémov
- Ako funguje grid computing?
- Pre aké typy úloh sa hodí grid computing?
- Charakterizujte cloud computing, jeho hlavné výhody a riziká
- Charakterizujte Internet of Things a jeho význam pre súčasný priemysel
- Vymenujte jednotlivé vrstvy hierarchického členenia Internet of Things
- Popíšte hlavné úlohy jednotlivých hierarchických vrstiev systému Internet of Things
- Vyjadrite hlavné trendy vo vývoji výpočtových modelov a dôsledky na HW, na návrh systémového SW a na sieťové technológie