😎😎😎 OMAD HAMMAGA!!!

**😎 LDEV 😎**

Laziz Jamoliddinov, [21.06.2023 11:46]

Strukturaviy dasturlash (yoki "tasvirchi dasturlash") dasturiy tizimlarni yaratishda, ulardagi kodning to'plamini o'rganishda yoki uni boshqarishda yordam beruvchi, kodni kelgusi yozuvchi yoki tahrirlovchi insonlar uchun, kodni o'qishni osonlashtiruvchi, loyihalashni yaxshilashga yordam beruvchi, xatolarni to'g'rilashga yordam beruvchi, va boshqa qulayliklarga ega bo'lish uchun tasviriy usullarni qo'llaydigan dasturiy dasturlashdir.

Strukturaviy dasturlash, dasturchilar va tizimni boshqaruvchilar uchun foydali bo'lishi mumkin. Ushbu usul tizimni tahlil qilishda va tizimni qurishda yordam beradi. Tasviriy dasturlash tizimni o'rganishda va uni boshqarishda juda ko'p yordam beradi. Strukturaviy dasturlashning eng mashhur dasturlash tillari, masalan, Visual Basic, C#, Java, Python, va Javascript kabi tillar hisoblanadi.

Strukturaviy dasturlash yordamida, dasturchilar va tizimni boshqaruvchilar tizimni o'rganish va uni boshqarishni osonlashtirishadi. Tasviriy dasturlash yordamida, kodlarni tahlil qilish, loyihalashni yaxshilash va xatolarni aniqlash va to'g'rilash osonlashadi. Bunda, tizimni boshqaruvchi va dasturchilar xato topish va uni to'g'rilashning vaqt va moliyaviy xarajatlardan qurtilishi mumkin.

Laziz Jamoliddinov, [21.06.2023 11:47]

Tizimli dasturlash (yoki "menejment dasturlash") bir qator vositalar va texnologiyalarni qo'llab-quvvatlash orqali tashkilotlarning biznes jarayonlarini boshqarish va avtomatlashtirish qobiliyatini oshirishni maqsad qiladigan IT sohasidir. Bu qobiliyatlar barcha sohalarda va tashkilot turlarida foydalanilishi mumkin, shuningdek, o'rta va katta kompaniyalarda, hukumat tashkilotlarida va boshqa tashkiliy strukturada ham foydalanilishi mumkin.

Tizimli dasturlash orqali tashkilotlar, masalan, xaridor-mijozlar bilan aloqalarini avtomatlashtirish, sotish, xarid qilish, hisob-kitob tizimlari, xodimlar maoshlarini hisoblash va boshqa ish jarayonlarini boshqarishni osonlashtirishlari mumkin. Tizimli dasturlash, tashkilotlarni ko'plab ishlarni avtomatlashtirish orqali xaridor-mijozlarga ko'plab xizmatlarni taqdim etishga imkon beradi va shu bilan birga, ish jarayonlarini optimallashtirishga imkon beradi.

Laziz Jamoliddinov, [21.06.2023 11:47]

Tizimli dasturlashni o'rganish uchun quyidagi tavsiyalar yordamida boshlang'ich qadamni olib borishingiz mumkin:

1. Asosiy dasturlash tillarini o'rganing: Tizimli dasturlash uchun asosiy dasturlash tillaridan bir yoki bir nechani o'rganing, masalan, Python, Java, C++, C# va boshqalar.

2. O'zingizni o'rganish platformalaridan foydalaning: Tizimli dasturlashni o'rganish uchun onlayn o'rganish platformalaridan foydalaning, masalan, Udemy, Coursera, edX va boshqalar.

3. O'zingizni o'rganish resurslaridan foydalaning: O'z-o'zini o'rganish uchun mavjud bo'lgan resurslardan, masalan, w3schools.com, codecademy.com, freecodecamp.org va boshqalaridan foydalaning.

4. O'zingizni o'rganish loyihasini boshlang: O'zingizni o'rganish loyihasini boshlang va uning ustida ishlashni boshlang. Bu sizga amaliy tajriba olishga imkon beradi va o'zingizni o'rganishni tezlashtiradi.

5. Kichik loyihalarda ishlash: Kichik tizimli dasturlash loyihalarida ishlash, masalan, tizimli dasturlash kurslarida yoki o'zingizni o'rganish uchun tuzilgan loyihalarda ishlay olasiz.

6. Yagona tizimli dasturlash platformalaridan foydalaning: Yagona tizimli dasturlash platformalaridan, masalan, Laravel, Django, Ruby on Rails, va boshqalardan foydalaning.

7. O'ziga qulay bo'lgan vositalardan foydalaning: Tizimli dasturlashni o'rganish uchun o'zingizga qulay bo'lgan vositalardan foydalaning, masalan, Visual Studio Code, Sublime Text, Atom va boshqalar.

8. O'zingizni o'rganish jamaatiga qo'shiling: O'zingizni o'rganish jamaatiga qo'shiling, masalan, Github, StackOverflow va boshqalar. Bu sizga tizimli dasturlash sohasida tajriba olishga imkon beradi va savollarga javob topishga yordam beradi.

Bu tavsiyalardan foydalanib, siz tizimli dasturlashni o'rganishga boshlash va o'zingizni professional darajada oshirishga qodir bo'lasiz.

Laziz Jamoliddinov, [21.06.2023 11:48]

Kodni hujjatlashtirish uchun quyidagi usullar keng tarqalgan:

1. Kommentariylar: Kodni hujjatlashtirish uchun eng oddiy usul kommentariylardir. Kommentariyalar kodning tushunchasini tushuntiradigan matnli qatorlar hisoblanadi. Bu qatorlar dasturchilar tomonidan kiritiladi va kodni qo'llash va tushuntirishga yordam beradi. Kommentariyalar kodni tahlil qilishda va yana ham muhim bo'lgan, kodni o'qish va tahrirlashni osonlashtiradi.

2. Dokumentatsiya: Kodni hujjatlashtirish uchun ikkinchi usul dasturiy tizimning dokumentatsiyasini yaratishdir. Bu odatda loyihalash jarayonida amalga oshiriladi va barcha tizimning funksiyalari, interfeyslari va boshqa xususiyatlari haqida ma'lumot beradi. Dokumentatsiya, dasturchilar va boshqa tizimni boshqaruvchilar uchun juda qulaydir, chunki buni qo'llash orqali tizimni tahlil qilish va uni boshqarish osonlashadi.

3. Kod tahriri: Kodni hujjatlashtirish uchun uchunchi usul kod tahriri yoki kodni formatlashdir. Kodni formatlash, kodni o'qishni osonlashtiradi va kodni tahlil qilishda yordam beradi. Formatlash kodning shakl va ko'rinishini tanlashga yordam beradi, masalan, bir qatorda necha belgidan foydalanish, o'rinli qatorlar, albatta joylashgan elementlarning shakl, va boshqa xususiyatlar.

4. Kodni yoritish: Kodni hujjatlashtirish uchun to'rtinchi usul kodni yoritishdir. Bu usul, kodni vizual ravishda yoritishga yordam beradi. Kodni yoritish, kodni o'qishni osonlashtiradi va kodni tahlil qilishda yana ham yordam beradi. Kodni yoritish quyidagi usullar orqali bajariladi, masalan, ranglar, xatcho'p, tablar, va boshqa vizual vositalar.

Bu usullar kodni hujjatlashtirishda qo'llaniladigan eng mashhur usullardir. Bu usullar dasturchilar va boshqa tizimni boshqaruvchilar uchun kodni o'qish va tahrirlashni osonlashtiradi va tizimni boshqarishni yaxshilashga yordam beradi.

Laziz Jamoliddinov, [21.06.2023 11:50]

Utilit (yoki "utilitalar") dasturiy vositalar hisoblanadi, bu vositalar odatda bir nechta muhim funksiyalarni bajarish uchun ishlatiladi. Bu utilitlar odatda o'ziga xos vazifalarni bajarish uchun maxsus tuzilmalar yoki dasturlar hisoblanadi.

Quyidagi utilit turlari mavjud:

1. System utilitalari: Operatsion tizimni boshqarish va uni o'chirish uchun ishlatilgan vositalar hisoblanadi. Masalan, Windows Task Manager, System Monitor, va boshqalar.

2. Disk utilitalari: Diskni formatlash, diskni qayta tiklash, diskni ta'mirlash va boshqa disk vazifalarini bajarish uchun ishlatiladi. Masalan, Disk Cleanup, Disk Defragmenter, va boshqalar.

3. Foydalanuvchi interfeysi utilitalari: Foydalanuvchi interfeysi bilan ishlashni osonlashtiradigan vositalar hisoblanadi. Masalan, Start Menu, Control Panel, va boshqalar.

4. Ma'lumotlar bazasi utilitalari: Ma'lumotlar bazasini boshqarish va ma'lumotlarni qayta tiklash uchun ishlatiladi. Masalan, MySQL, Oracle, va boshqalar.

5. Qo'llab-quvvatlash utilitalari: Qo'llab-quvvatlash ishlarini bajarish uchun ishlatiladi. Masalan, Remote Desktop, TeamViewer, va boshqalar.

6. Tarmoq utilitalari: Tarmoqni boshqarish va tarmoq xatolarni aniqlash uchun ishlatiladi. Masalan, Wireshark, Network Monitor, va boshqalar.

7. Multimedia utilitalari: Multimedia fayllarni ijro etish va tahrirlash uchun ishlatiladi. Masalan, Windows Media Player, VLC Media Player, va boshqalar.

8. Ofis utilitalari: Ofis vazifalarini bajarish uchun ishlatiladi. Masalan, Microsoft Office, OpenOffice, va boshqalar.

Bu utilitlar turli sohalarda foydalaniladi va ulardan foydalanish, dasturchilar va tizimni boshqaruvchilar uchun juda qulaydir. Utilitalar, tizimni boshqarishni va tizimni yaxshilashni osonlashtiradi va tizimning ishga tushirish jarayonlarini optimallashtiradi.

Laziz Jamoliddinov, [21.06.2023 11:51]

Translyator, interpretator va kompilyator dasturlar kodni bajargan dasturchilar uchun juda muhim bo'lgan dasturiy vositalardir. Ularning har birining o'ziga xos xususiyatlari va qulayliklari mavjud.

1. Translyator dasturlari: Bu dasturlar, bir dastur tili yozilganligi va uni boshqa tilga tarjima qilishga yordam beradi. Bu tarjima amalga oshirilgan kodni o'zgartirmaydi, balki uni boshqa tilda yozilgan kodga o'giradi. Translyator dasturlari quyidagi turlar bo'lishi mumkin:

- Source-to-source translyatorlar: Bir dastur tili yozilganligi va uni boshqa tilga tarjima qilish uchun ishlatiladi. Masalan, Google Translate API.

- Binary-to-text translyatorlar: Bunday dasturlar, ikki tarjima qilinadigan formatni (masalan, binar va matnli) o'zaro almashtiradigan dasturlardir. Masalan, Base64 encoding/decoding dasturlari.

2. Interpretator dasturlari: Bu dasturlar, kodni bajarish jarayonida, kodni ko'chirish, tahlil qilish va bajarishni bajarishga yordam beradi. Interpretator dasturlari, kodni javobga aylanishi davomida qo'llaniladi va kodni faqat bir marta bajargan holda bajargan holda yoritadi. Bu dasturlar quyidagi turlar bo'lishi mumkin:

- Python, Ruby, Perl, Shell va boshqa skriptlash tilidagi interpretator dasturlar.

- JavaScript dasturlari, brauzer yordamida interpretatsiya qilinadi.

3. Kompilyator dasturlari: Bu dasturlar, kodni bajarish jarayonida, kodni tahlil qilish va uni bajarish uchun bitta yoki bir nechta fayl yaratadi. Kompilyator dasturiy tizimlarning ishga tushirilishi paytida kodni o'qimasdan bajaradi, shuning uchun uni bir marta bajargan holda yoritadi. Bu dasturlar quyidagi turlar bo'lishi mumkin:

- C++, Java, C# va boshqa kompilyatsiya qiluvchi dasturlar.

- HTML va CSS, brauzer yordamida interpretatsiya qilinadi.

Barcha usullar, dasturchilar uchun qulayliklar taqdim etadi va ulardan foydalanish, kodni tahlil qilish va uni bajarishni osonlashtiradi. Barcha usullar o'ziga xos xususiyatlarga ega bo'lishi mumkin va foydalanuvchilar uchun qulayliklar taqdim etadi.

Laziz Jamoliddinov, [21.06.2023 11:52]

Aloqa muharrirlari tizimni qayta ishlovchi dasturlar quyidagi turlarni o'z ichiga oladi:

1. Ping: Bu dastur, tarmoq qurilmalariga murojaat qilish va ularga qo'shimcha ma'lumotlar olish uchun ishlatiladi. Ping dasturi bir IP manziliga yoki nomiga ping so'rovini yuboradi va tarmoq qurilmasining javobini olishga harakat qiladi. Bu dastur tarmoq qurilmasining holatini aniqlash uchun juda qulaydir.

2. Traceroute: Bu dastur, tarmoq qurilmasiga yo'lning qaysi qismida xatolik borligini aniqlash uchun ishlatiladi. Traceroute dasturi, yo'lning har bir qismida o'tilgan vaqt, IP manzillari va tarmoq qurilmasining nomi kabi ma'lumotlarni ko'rsatadi.

3. Netstat: Bu dastur, tarmoq qurilmasidagi portlarga qaysi tarmoq qurilmalarining ulanishi va qanday maqsadlar uchun ulanishini ko'rsatadi. Netstat dasturi, tarmoq qurilmasining o'tish vaqti, port raqami, IP manzillari va tarmoq qurilmasining nomi kabi ma'lumotlarni ko'rsatadi.

4. Nmap: Bu dastur, tarmoq qurilmasidagi tizimlar va ularga ulanish portlari haqida ma'lumot olish uchun ishlatiladi. Nmap dasturi, tarmoq qurilmasining portlari, ulanish yoki tizimlarining nomi va versiyalari kabi ma'lumotlarni ko'rsatadi.

5. Wireshark: Bu dastur, tarmoqda jo'natilgan paketlarni taqqoslash va analiz qilish uchun ishlatiladi. Wireshark dasturi, tarmoqda jo'natilgan paketlarni o'qish, yozish va tahlil qilishga yordam beradi.

6. Tcpdump: Bu dastur, tarmoqda jo'natilgan paketlarni o'qish va ularga qo'shimcha ma'lumotlar olish uchun ishlatiladi. Tcpdump dasturi, tarmoqda jo'natilgan paketlarni o'qish, yozish va tahlil qilishga yordam beradi.

Bular aloqa muharrirlari uchun juda muhim dasturlar hisoblanadi, chunki ulardan foydalanish, tarmoq muhitini tahlil qilish va yana ham muhim, tarmoq qurilmasidagi xatoliklarni aniqlashni osonlashtiradi.

Laziz Jamoliddinov, [21.06.2023 11:53]

Yuklovchi dasturlar kompyuterga yuklanadigan fayllarni yuklash uchun ishlatiladi. Otladchik dasturlar esa dasturlashda yoki tizimni ishga tushirishda xatolikni topish va tuzatish uchun ishlatiladi. Dasturlashda va tizimni ishga tushirishda ko'plab xatolar yuzaga kelishi mumkin, va bu xatoliklarni aniqlash va tuzatish uchun otladchik dasturlar foydalaniladi. Quyidagi misollar yuklovchi va otladchik dasturlarga misol ko'rsatadi:

1. Yuklovchi dasturlar: Google Chrome, Mozilla Firefox, Microsoft Edge, Adobe Acrobat Reader, Dropbox, uTorrent, Skype, Microsoft Office, Adobe Creative Cloud, VLC Media Player, WinRAR, Telegram, WhatsApp, Adobe Photoshop, Visual Studio, Android Studio, Xcode, iTunes, Spotify, Steam, va hokazo.

2. Otladchik dasturlar: Visual Studio Code, Eclipse, IntelliJ IDEA, Android Studio, Xcode, PyCharm, NetBeans, Code::Blocks, Sublime Text, Atom, WebStorm, gdb, Valgrind, strace, ltrace, Wireshark, Fiddler, Postman, Charles, va hokazo.

Yuklovchi dasturlar har bir kompyuter muhitida juda ko'p ishlatiladi, chunki ular foydalanuvchilar tomonidan ko'p ishda qo'llaniladigan dasturlardir. Otladchik dasturlar esa dasturlash va tizimni ishga tushirishda xatolikni topish va tuzatish uchun juda muhim dasturlardir. Otladchik dasturlar sizga dasturlashda yoki tizimni ishga tushirishda yuzaga kelgan xatoliklarni tuzatishga yordam beradi va shuningdek dastur yoki tizimni to'liq tahlil qilishga imkon beradi.

Laziz Jamoliddinov, [21.06.2023 11:54]

Fayl tizimlari, operatsion tizim klassifikatsiyasi, dasturlash tillarida funksiyalar va dasturlash tillarida metodlar quyidagi tarzda ta'riflanadi:

1. Fayl tizimlari: Fayl tizimi, fayllarni saqlash va tahlil qilish uchun ishlatiluvchi tizimdir. Fayllar tizimi, fayllarni ma'lum bir ko'rsatkichga saqlash va ularni o'qish, yozish, ko'chirish, o'chirish va tahrirlashni osonlashtiradi. Ba'zi fayl tizimlari quyidagi kabi xususiyatlarga ega bo'lishi mumkin: ma'lumotlarni to'plash, ma'lumotlar bazasi, ma'lumotlar ombori, fayllar tarmoqi va hokazo.

2. Operatsion tizim klassifikatsiyasi: Operatsion tizim, kompyuterda ishlaydigan barcha dasturlarni boshqarish, boshqarish va ishga tushirish uchun ishlatiluvchi tizimdir. Operatsion tizimlar quyidagi turdagi tizimlar bo'lishi mumkin: Windows, MacOS, Linux, iOS, Android, va hokazo.

3. Dasturlash tillarida funksiyalar: Dasturlash tillarida funksiya, belgilangan vazifani bajarish uchun yozilgan moduldur. Funksiyalar yordamida, biror vazifani bajarish uchun kerakli kodni yozish mumkin. Funksiyalar, dasturlash tillarida qo'llaniladigan asosiy modullardan biridir va ularga nom, parametrlar va qaytuvchi qiymat beriladi.

4. Dasturlash tillarida metodlar: Dasturlash tillarida metod, obyektga tegishli vazifalarni bajarish uchun yozilgan funksiyalardir. Metodlar, obyektga tegishli vazifalarni bajarish uchun kerakli kodni osonlashtiradi. Metodlar, dasturlash tillarida obyektoriyentlangan dasturlashining asosiy qismidir va ularga obyekt nomi, parametrlar va qaytuvchi qiymat beriladi.

Laziz Jamoliddinov, [21.06.2023 11:55]

1. DLL kutubxonalarining asosiy maqsadi: DLL (Dynamic Link Library) kutubxonalar, bir nechta dasturlar tomonidan birlashgan va o'zaro almashinuvni ta'minlaydigan fayllardir. DLL kutubxonalar, bitta dasturda yozilgan kodni boshqa dasturlarga osonlik bilan qo'llash imkonini beradi. Bunday fayllar avvalgilar bilan qiyoslash va ulardan foydalanish imkonini beradi. DLL kutubxonalarining asosiy maqsadi esa, kodni doimiy tarzda qayta ishlatish va kodni qayta ishlatish vaqtidan tejashdir.

2. Tizimli dasturlash: Tizimli dasturlash, kompyuter tarmoq tizimlarini va operatsion tizimlarni boshqarishga mo'ljallangan dasturlash turi. Tizimli dasturlash, dasturlash tillari va asosiy kutubxonalar yordamida operatsion tizimlarni tuzish, tarmoq tizimlarini yaratish va ulardagi xatoliklarni aniqlash va tuzish kabi vazifalarini bajaradi. Tizimli dasturlash, kompyuter tarmoqlari va ularda yaratilgan tizimlar orqali ma'lumot almashish, kompyuterlarning birlashmasini ta'minlash, ma'lumotlarni saqlash va ulardan foydalanish kabi vazifalarni bajarish uchun juda muhimdir. Tizimli dasturlash, operatsion tizimlar va tarmoq tizimlari kabi katta tizimlarni boshqarish uchun yuqori darajada uskunalar yaratishga imkon beradi.

Laziz Jamoliddinov, [21.06.2023 12:07]

1. Tizimli dasturlashning asosiy xususiyatlari: Tizimli dasturlash, kompyuter tarmoq tizimlarini va operatsion tizimlarni boshqarishga mo'ljallangan dasturlash turi. Tizimli dasturlashning asosiy xususiyatlari quyidagilardan iborat bo'ladi:

- Tizimli dasturlashda, dasturlash tillari va asosiy kutubxonalar yordamida operatsion tizimlarni tuzish, tarmoq tizimlarini yaratish va ulardagi xatoliklarni aniqlash va tuzish kabi vazifalarini bajarish mumkin.

- Tizimli dasturlash, operatsion tizimlar va tarmoq tizimlari kabi katta tizimlarni boshqarish uchun yuqori darajada uskunalar yaratishga imkon beradi.

- Tizimli dasturlashda, dastur yozilish shakli va modullar o'zgarishi yuzaga kelishi mumkinligi tufayli, kodni o'zgartirish va qo'shish osonlik bilan amalga oshiriladi.

2. Tizimli dasturlash amaliy dasturlashdan farqlari: Tizimli dasturlash amaliy dasturlashdan farqli bo'lib, amaliy dasturlash yuqori darajada o'ziga xos maqsadlarga mo'ljallangan va oddiy dasturlash vazifalarini bajarishga mo'ljallangan dasturlash turidir. Amaliy dasturlashda, qo'shimcha modullar va kutubxonalar ishlatilmaydi va kod yozilish shakli oddiyroqdir. Tizimli dasturlash esa, operatsion tizimlarni va tarmoq tizimlarini boshqarish va ulardagi xatoliklarni aniqlash va tuzish kabi vazifalarini bajarish uchun yozilgan dasturlash turiga o'xshashlikka ega, ammo unga xos xususiyatlari mavjud.

3. Tizimli dasturlashda qanday dasturlash tillari keng qo'llaniladi: Tizimli dasturlashda, dasturlash tillari va asosiy kutubxonalar yordamida operatsion tizimlarni tuzish, tarmoq tizimlarini yaratish va ulardagi xatoliklarni aniqlash va tuzish kabi vazifalarini bajarish uchun quyidagi dasturlash tillari keng qo'llaniladi:

- C va C++ dasturlash tillari: Tizimli dasturlashning asosiy dasturlash tillaridan biri C va C++ dasturlash tillari hisoblanadi. Bu tillar, tizimli dasturlashda yuqori darajada ishlatiladi, chunki ular xavfsiz va tez ishlaydigan kod yozish imkonini beradi.

- Java dasturlash tili: Java, tizimli dasturlashda amaliyotga kiritilgan tizimlarni boshqarish va tarmoq tizimlarini yaratish uchun ham ishlatiladi. Java Virtual Machine (JVM) yordamida, Java dasturlari ko'p platformalarda to'g'ridan-to'g'ri ishlaydi.

- Python dasturlash tili: Python, tizimli dasturlashda ham foydalaniladigan dasturlash tillari jihatidan qo'llaniladi. Python, qulay sintaksis va modullarining kengligi sababli, tizimli dasturlash sohasida ham ommalashtiriladi.

4. Tizim chaqiruvlari nima va ular nima uchun kerak: Tizim chaqiruvlari,dasturlash tilida tizimli dastur va funksiyalarga o'xshashlikka ega bo'lgan chaqiruvlardir. Ular dasturlash tili yordamida operatsion tizimlar va tarmoq tizimlarini boshqarishda ishlatiladi. Tizim chaqiruvlari barcha tizimli dasturlash tillarida mavjud bo'lib, ularga nom va parametrlar beriladi. Tizim chaqiruvlari, tizimli dasturlashda bir nechta asosiy maqsadlarga mo'ljallangan, masalan:

- Operatsion tizimlardan ma'lumot olish va ularga ma'lumot jo'natish;

- Tizimlar orasidagi almashishni boshqarish;

- Tarmoq tizimlarini boshqarish, ma'lumot almashish va ularga ma'lumot jo'natish;

- Foydalanuvchidan ma'lumot olish va ularga javob bering;

- Tizimlardagi xatoliklarni aniqlash va ulardan to'g'ridan-to'g'ri foydalanish.

Tizim chaqiruvlari, operatsion tizimlarni va tarmoq tizimlarini boshqarishda juda muhimdir, chunki ularga qayta qayta qo'shilish va ulardan foydalanish mumkinligi mavjud.

Laziz Jamoliddinov, [21.06.2023 12:09]

Operatsion tizimlar tizimni dasturlash uchun quyidagi funksiyalarni beradi:

1. Fayl tizimi: Fayl tizimi, operatsion tizimlarining fayllar va direktoriyalarni boshqarish uchun ishlatiladigan funksiyalardan biridir. Bu funksiyalar, fayllarni yaratish, ko'chirish, o'chirish, nomini o'zgartirish va fayllar orasida ko'chirish kabi amallarni bajarish imkonini beradi.

2. Protsesslar: Operatsion tizimlar, bitta yoki bir nechta dasturlarni bajarish uchun ishlatiladigan protsesslar yaratishga imkon beradi. Bu protsesslar, operatsion tizimlarining harakatlarini bajarish uchun muhimdir.

3. Tarmoq tizimi: Operatsion tizimlar, tarmoq tizimlarini boshqarish uchun imkonlar ta'minlayadi. Bu funksiyalar, tarmoq tizimlarini yaratish, ularga ulanish va ulardan ma'lumot almashish kabi amallarni bajarish imkonini beradi.

4. Xatoliklar va ko'rsatkichlar: Operatsion tizimlar, xatoliklar va ko'rsatkichlar yaratish uchun imkonlar ta'minlayadi. Bu funksiyalar, xatoliklarni aniqlash, ulardan xabarnoma olish va ulardan foydalanish kabi vazifalar uchun ishlatiladi.

5. Ruxsatnoma boshqaruvchi: Operatsion tizimlar, foydalanuvchilar va dasturlar orasidagi ruxsatnomalar uchun imkonlar ta'minlayadi. Bu funksiyalar, fayllar va direktoriyalarga kirish ruxsatnomalarini boshqarish, foydalanuvchilar uchun ma'lumotlarni yashirish va ularga kirish ruxsatnomasi berish kabi amallarni bajarish imkonini beradi.

6. Interfeyslar: Operatsion tizimlar, foydalanuvchilar uchun interfeyslar yaratishga imkon beradi. Bu funksiyalar, foydalanuvchilar uchun ikki-uchta yoki ko'p elementdagi oynalar va ularga qo'shimcha xususiyatlar qo'shish kabi vazifalar uchun ishlatiladi.

Bu funksiyalar, operatsion tizimlarining tizimni boshqarishda keng qo'llaniladigan funksiyalardan faqat bir qismidir. Boshqa funksiyalar ham mavjud va ular operatsion tizimlarning turiga va versiyasiga qarab o'zgaradi.

Laziz Jamoliddinov, [21.06.2023 12:11]

1. Tizimli dasturlash kompyuter texnikasi bilan bog'liq: Tizimli dasturlash, kompyuter texnikasi bilan eng yaqin bog'liq turlardan biridir. Tizimli dasturlash, kompyuter tizimlarini boshqarish, tarmoq tizimlarini yaratish va ulardagi xatoliklarni aniqlash va tuzish kabi vazifalarini bajarishda muhim hisoblanadi. Kompyuter tizimlari esa, tizimli dasturlash yordamida yaratilgan dasturlar orqali ishlaydi. Bunday qilib, tizimli dasturlash va kompyuter tizimi bir-biriga qarshi-katta bo'lishmasligi mumkin.

2. Tizimli dasturlash kompyuter tizimining ishlashiga ta'sir qiladi: Tizimli dasturlash, kompyuter tizimining ishlashiga katta ta'sir qiladi. Tizimli dasturlash yordamida yaratilgan dasturlar, kompyuter tizimining barcha hissalarini boshqarishga imkon beradi. Bu dasturlar, operatsion tizimlarni yaratish, tarmoq tizimlarini boshqarish, fayllar va direktoriyalarni boshqarish, ko'rsatkichlar va xatoliklar yaratish, foydalanuvchilar uchun interfeyslar yaratish kabi muhim vazifalarni bajarganholda, tizimning ishlashida muhim rol o'ynaydi.

3. Tizim dasturiy ta'minotini ishlab chiqishda muammolar: Tizim dasturiy ta'minotini ishlab chiqishda bir nechta muammolar paydo bo'lishi mumkin. Ba'zilaridan ular quyidagilardir:

- Xatoliklar yaratish: Tizim dasturlarining yaratilishi davomida xatoliklar paydo bo'lishi mumkin. Bu xatoliklar, kodning noto'g'ri yozilishi, dasturni tekshirish va test qilishda yashirin xatoliklar kabi sabablarga qarab yuzaga kelishi mumkin.

- Yomon kod yozish: Tizim dasturlarini yaratishda yomon kod yozish muammolari paydo bo'lishi mumkin. Yomon kod yozish, dasturning ishlashini buzish, kompyuter tizimining ishlashida xatoliklar yaratish kabi muhim muammolarga olib kelishi mumkin.

- Qulaylik va samarali dasturlar yaratish: Tizim dasturiy ta'minotini ishlab chiqishda, foydalanuvchilarga qulay va samarali dasturlar yaratish muhimdir. Foydalanuvchilar, dasturlarning qulay va samarali bo'lishini kutishlari mumkin, shuning uchun dasturlarni ularga mos ravishda ishlab chiqish kerak.

- Muhim tizim chaqiruvlari yaratish: Tizim dasturiy ta'minotini ishlab chiqishda muhim tizim chaqiruvlari yaratish lozim. Tizim chaqiruvlari, operatsion tizimlarni va tarmoq tizimlarini boshqarishda yordam beradi, shuning uchun ularga katta e'tibor bermoq kerak.

Laziz Jamoliddinov, [21.06.2023 12:12]

Kodni tuzishda konseptlardan foydalanish muhimdir, chunki bu usul kodni yangilash va o'zgartirishda yordam beradi. Quyidagi konseptlardan foydalanish tavsiya etiladi:

1. DRY (Don't Repeat Yourself): Bu konsept kodni ko'proq qayta yozishdan saqlaydi. Ma'nosi shundaki, biror bir vazifani bajarish uchun bir nechta joyda kodni yozishdan buziladi. Bu esa kodni o'zgartirish va yangilashda qiyinchilik yaratadi.

2. SOLID (Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion): Bu konsept, ob'ektoriy tuzilma yordamida kodni tuzishda ishlatiladi. Bunday qilib, ob'ektlar biror bir vazifani bajarish uchun yaratiladi va har bir ob'ekt yagona vazifani bajarishi uchun javob beradi. Bu esa kodni tuzishda o'zgarishlarga va yangiliklarga qarshi osonlik ta'minlaydi.

3. KISS (Keep It Simple, Stupid): Bu konsept kodni oddiy va tushunarli qilishga yordam beradi. Kod ko'proq vaqtni va kuchni sarflamaydi, chunki kodni tushunarli va oson bo'lishi lozim.

4. YAGNI (You Ain't Gonna Need It): Bu konsept, kodga juda ko'p funksiyalar qo'shishni to'xtatadi. Maqsadi, o'zgarishlar yaratish va yangiliklarga qarshi qo'pollikni o'z ichiga oladi.

5. TDD (Test-Driven Development): Bu konsept kodni test qilishni o'zi tuzishda yordam beradi. Testlar kodni muvaffaqiyatli yozishga yordam beradi va kodni yangilashda yordam beradi.

Bu konseptlar, kodni tuzishda yordam beradi va kodni o'zgartirish va yangilashda qiyinlik tugatadi. Shuningdek, kodni tuzishda muhim bo'lgan boshqa konseptlar ham mavjud, shu jumladan DRY, YAGNI va KISS konseptlariga qo'shimcha ko'rsatmalar.

Laziz Jamoliddinov, [21.06.2023 12:13]

Ob'yekt va klasslar, ob'ektoriy tuzilma asosida kod tuzilishida eng muhim elementlar hisoblanadi. Ob'yekt va klasslarning farqi quyidagilardir:

1. Klass, ob'ektlarning tavsifini ko'rsatadi. Klass, ob'ektlarning yaratilishiga yordam beradi va ob'ektlarning tavsifini ko'rsatadi. Boshqa so'zlar bilan, klass, ob'ektlar yaratishda ishlatiladigan bir turlidagi chiziqchi hisoblanadi.

2. Ob'yekt, klasslarga asoslanadi. Ob'yekt, klassning yaratilgan nusxasi hisoblanadi. Klass, ob'ektlarni yaratishga yordam beradi, ob'yekt esa klassning nusxasi sifatida aniqlanadi.

3. Klass, o'zgartirishsiz bo'lishi mumkin. Klass, ob'ektlarning asos qismlarini aniqlaydi, ammo o'zgartirishsiz bo'lishi mumkin. Bunday qilib, klass, ob'ektlarning asosiy tavsiflarini ko'rsatadi, ammo u ob'ektlarning qiymatlarini o'zgartirishga imkon bermaydi.

4. Ob'yekt, o'zgartirishsiz bo'lishmaydi. Ob'yekt, klassning nusxasi hisoblanadi va o'zgartirishsiz bo'lishmaydi. Ob'yekt, klassning qiymatlarini o'zgartirishga imkon bermaydi, ammo u klassning tavsiflarini saqlaydi.

5. Klass, ob'ektlarning shablonini ko'rsatadi. Klass, ob'ektlarning shablonini ko'rsatadi, ammo ob'yekt o'ziga xos tavsiflarga ega bo'lgan klassning nusxasi hisoblanadi.

6. Ob'yekt, klassning nusxasi hisoblanadi. Ob'yekt, klassning nusxasi hisoblanadi va klass ob'yektlarning yaratilishiga yordam beradi.

Bunday qilib, klass va ob'yekt, ob'ektoriy tuzilma asosida kod tuzishda muhim elementlar hisoblanadi. Klass, ob'ektlarning tavsifini ko'rsatadi va ob'ektlar yaratishda ishlatiladi, ob'yekt esa klassning nusxasi sifatida aniqlanadi.

Laziz Jamoliddinov, [21.06.2023 12:14]

1. Tizimli dasturlashda tizim kutubxonalarining roli: Tizim kutubxonalarining roli tizimli dasturlashda muhimdir. Tizim kutubxonalar, bir nechta foydali funksiyalarni tizimli dasturlashda ishlatishga imkon beradi. Bunday qilib, tizim kutubxonalar, tizim dasturlari tuzishda va ularga qo'shimcha funksiyalar qo'shishda yordam beradi.

2. C++ dasturlash tilida qanday umumiy kutubxonalar ishlatiladi: C++ dasturlash tilida umumiy kutubxonalar ishlatilishi odatda quyidagi tartibda amalga oshiriladi:

- #include direktivasi yordamida, umumiy kutubxonalar kode yozilgan faylga qo'shiladi.

- Umumiy kutubxonalar yordamida, qo'shimcha funksiyalar yaratiladi va ularga o'zgartirishlar kiritiladi.

- Umumiy kutubxonalar, yana bir necha kutubxonalar bilan birlikda ishlatilishi mumkin.

3. Qurilma drayveri nima va u operasion tizim hamda apparat qurilmalari o‘rtasidagi aloqani qanday ta’minlaydi: Qurilma drayveri, operatsion tizimga qurilma qurilishining qo'llanilishi uchun zarur bo'lgan ma'lumotlarni ta'minlaydi. Qurilma drayverlari, qurilma va operatsion tizimi o'rtasidagi aloqani boshqarish uchun xizmat qiladi. Qurilma drayverlari, qurilma qurilishining barcha tizimlarini boshqarishda muhim hisoblanadi.

Qurilma drayverlari, operatsion tizim va qurilma qurilishlari o'rtasidagi aloqani ta'minlashda muhim rol o'ynaydi. Qurilma drayverlari, operatsion tizimga qurilma qurilishining qo'llanilishi uchun kerakli ma'lumotlarni ta'minlashda yordam beradi. Bunday qilib, qurilma drayverlari, operatsion tizim va qurilma qurilishlari o'rtasidagi aloqani ta'minlashda muhim hisoblanadi.

Laziz Jamoliddinov, [21.06.2023 12:15]

Tizimli dasturlar, kompyuter tizimlarini boshqarish uchun tuzilgan dasturlardir. Bunday tizimlar, bitta yoki bir nechta tizimli dasturlaridan iborat bo'lishi mumkin. Tizimli dasturlar, foydalanuvchilarning ko'rsatmalari asosida ishlaydi va odatda operatsion tizimlarida ishlatiladi.

Uskunaviy dasturiy ta'minotlar, tizimlarni himoyalash va muhofaza qilish uchun ishlatiladigan dasturlardir. Bunday ta'minotlar, viruslardan, hakerlardan va boshqa xavfli elementlardan tizimlarni himoyalashda muhim bo'lib, odatda antivirys dasturlari, tuzatuvchi dasturlar va boshqa xavfsizlik dasturlaridan iborat bo'lishi mumkin.

Amaliy dasturlar, odatda foydalanuvchilarning kundalik hayoti yoki biznes faoliyati uchun ishlatiladigan dasturlardir. Bunday dasturlar, foydalanuvchilarga xizmat qilish, sotuvlar va xaridorlar bazasini boshqarish, hisob-kitoblarni yuritish va boshqa faoliyatlar uchun ishlatiladi. Amaliy dasturlar, odatda foydalanuvchilarga o'zlarining talablari asosida yaratiladi va foydalanuvchilarning to'g'ridan-to'g'ri talablari bajarishiga yordam beradi.

Barcha ular, tizimni yaxshi ishga tushirish uchun muhim hisoblanadi va sodda yoki murakkab muammolarni hal qilishda yordam beradi.

Laziz Jamoliddinov, [21.06.2023 12:16]

SHK (Sanoat hisobot kartasi) - bu korporativ dasturiy tizim, sanoat sohasida ishlovchi tashkilotlar o'rtasida yozilgan tashkiliy va hisobotlash dasturlari tizimini ta'minlash uchun ishlab chiqilgan. SHK tizimi, sanoat tizimlarining barcha ko'rinishlarida ishlatiladi va tarkibida qo'shimcha modullar va qo'shimcha funksiyalar bor.

SHK konfiguratsiyasi, SHK tizimining sozlamalarini, o'ziga xos tavsiflarini va qo'llanishni aniqlash uchun kerakli ma'lumotlarni o'z ichiga oladi. Bunday qilib, SHK konfiguratsiyasi, tizimga o'ziga xos tavsiflar kiritishga yordam beradi va tizimni shaxsiy talablarga moslashtirishga imkon beradi.

SHK asosiy platasi va qurilmalari, SHK tizimi uchun kerakli qurilmalardan tashkil topgan. SHK asosiy platasi, tizimni boshqarish uchun kerakli asosiy komponentlardan iborat bo'lib, barcha yordamchi funksiyalarni o'z ichiga oladi. SHK qurilmalari esa, asosiy platformaga qo'shimcha qurilmalar hisoblanadi. SHK qurilmalari, tizimga qo'shimcha funksiyalar qo'shish uchun ishlatiladi va tizimni kengaytirishga imkon beradi.

SHK tizimi, tarkibida SHK Server, SHK Klient, SHK Qo'shimcha qurilmalar, SHK O'zgartirishlar va SHK Hisobotlar kabi qurilmalarni o'z ichiga oladi. SHK Server, tizimni boshqarish uchun kerakli asosiy qurilma hisoblanadi. SHK Klient esa, SHK Server bilan aloqani ta'minlaydi va foydalanuvchilarga tizimda ishlash imkonini beradi. SHK Qo'shimcha qurilmalari, SHK tizimini kengaytirish uchun qo'shimcha qurilmalardan iborat bo'lib, SHK O'zgartirishlar esa, tizimni o'zgartirishlar uchun ishlatiladi. SHK Hisobotlar esa, tizimning ma'lumotlar bazasidan statistik ma'lumotlar va hisobotlar olish uchun ishlatiladi.

Bunday qilib, SHK tizimi, sanoat sohasida ishlovchi tashkilotlar o'rtasida yozilgan tashkiliy va hisobotlash dasturlari tizimini ta'minlash uchun ishlab chiqilgan va tarkibida SHK Server, SHK Klient, SHK Qo'shimcha qurilmalar, SHK O'zgartirishlar va SHK Hisobotlar kabi qurilmalar bor.

Laziz Jamoliddinov, [21.06.2023 12:17]

Kompyuter dasturiy ta'minotning (DT) tuzilishi, kompyuter dasturlarini boshqarish va ishlatish uchun kerakli funksiyalarni, ma'lumotlarni va resurslarni ta'minlashga yordam beradi. DT tuzilishi, kerakli ma'lumotlar va resurslar tashkil qilinishi, yozilgan kodning tushunarli, to'g'ri va ishga tushirilishi, tizimlar orasidagi aloqalarni boshqarish uchun yordam beradigan funksiyalar yaratilishi yoki ishlatilishi, ma'lumotlar bazasi boshqarish, audiotarjimaga yordam berish, xavfsizlik va himoya va boshqa funksiyalarni o'z ichiga oladi.

Kompyuter dasturiy ta'minotning (DT) turlari odatda quyidagi tartibda bo'lishi mumkin:

1. Tizimli dastur: bu turi dasturlar, operatsion tizimlarni boshqarish, tizimlarni boshqarish va tizimlarni boshqarish uchun ishlatiladigan ma'lumotlar bazalarini boshqarish uchun ishlatiladi.

2. Amaliy dastur: bu turi dasturlar, odatda foydalanuvchilarning kundalik hayoti yoki biznes faoliyati uchun ishlatiladigan dasturlardir. Bunday dasturlar, foydalanuvchilarga xizmat qilish, sotuvlar va xaridorlar bazasini boshqarish, hisob-kitoblarni yuritish va boshqa faoliyatlar uchun ishlatiladi.

3. Utilitalar: bu turi dasturlar, foydalanuvchilarning kompyuterlarida amalga oshiriladigan sodda va murakkab muammolarni hal qilish uchun qo'llaniladi. Bunday dasturlar, fayllar va papkalarni kengaytirish, ma'lumotlarni nusxalash, qaytarish va boshqa amallarni bajarish uchun yordam beradi.

Kompyuter dasturiy ta'minotning (DT) tuzilishi va turlari, kompyuter dasturlarini boshqarish va ishlatish uchun kerakli funksiyalarni, ma'lumotlarni va resurslarni ta'minlashga yordam beradi. DT turlari odatda tizimli dastur, amaliy dastur va utilitalar shakllarida bo'lib, ular har biri o'ziga xos vazifalarni bajaradi.

Laziz Jamoliddinov, [21.06.2023 12:18]

Programmalash tillari, dasturlar va tizimlar yaratish uchun ishlatiladigan qismlar, funksiyalar va boshqa vositalardir. Bu tillar, yozilgan kodni tushunarli va ishga tushirilishi uchun kerakli imkoniyatlar, funksiyalar va o'zgaruvchilar kabi qo'llanishlarni taklif qiladi. Programmalash tillari, dasturlash tillari yoki programmalash tillari sifatida ham ataladi.

Programmalash tillari turli xil bo'lishi mumkin, quyidagi misollar keltirilishi mumkin:

1. C++ - bu turli xil platformalarda ishlaydigan, yuqori darajada yuqori darajada, yuqori darajada bo'lgan dasturlash tili.

2. Java - bu, platformaga moslashtirilgan xususiyatlari va ko'pgina imkoniyatlari bilan xarakterlangan, platformaga moslashtirilgan dasturlash tili.

3. Python - bu, sodda sintaksisga ega bo'lgan va murakkab kompyuter dasturlari yaratishga imkon beradigan dasturlash tili.

4. PHP - bu, veb-saytlar yaratish uchun ishlatiladigan dasturlash tilidir.

Programmalash tillari turlari, dasturlar va tizimlar yaratish uchun kerakli imkoniyatlar, funksiyalar va o'zgaruvchilar kabi qo'llanishlarni taklif qiladi. Bu turlar odatda quyidagi tartibda bo'lishi mumkin:

1. Tizimli dasturlash tillari - bu til, operatsion tizimlarni boshqarish va tizimlarni boshqarish uchun ishlatiladigan ma'lumotlar bazalarini boshqarish uchun ishlatiladi.

2. Obyektga yo'naltirilgan dasturlash tillari - bu til, dasturlashda obyektlar bilan ishlashga imkon beradi.

3. Skript dasturlash tillari - bu til, murakkab kompyuter dasturlari yaratish uchun sodda sintaksis bilan ishlovchi til.

Programmalashda instrumental vositalar, dasturlar va tizimlarni yaratish uchun ishlatiladigan vositalardir. Bu vositalar, programmalash tillariga qo'shimcha imkoniyatlar, funksiyalar va o'zgaruvchilar kabi taklif qiladi. Muhim instrumental vositalar misollaridan ba'zi, Visual Studio, Eclipse, NetBeans va IntelliJ IDEA ni kiritish mumkin.

Laziz Jamoliddinov, [21.06.2023 12:19]

Operatsion tizim (OS), kompyuter tizimida ishlovchi dasturlarni boshqarish, tizim resurslarini ta'minlash, fayllarni boshqarish, aloqa protokollari yordamida tarmoqqa kirish, xavfsizlik va himoyani ta'minlash va boshqa funksiyalarni bajarish uchun kerakli vositalarni ta'minlash uchun yaratilgan tizimdir. OS, kompyuterda barcha boshqa dasturlar va tizimlar uchun asosiy tizim hisoblanadi.

OS tuzilishi quyidagi qismatlaridan tashkil topgan:

1. Yadro (Kernel): Bu OS tizimining asosiy qismi hisoblanadi va barcha tizim resurslarini boshqaradi. Bu yadro, kompyuter tizimida ishlovchi dasturlar bilan aloqada bo'lish uchun kerakli protokollarni yuritadi.

2. Tarmoq protokollari va qurilmalar: Bu qismat, tarmoqqa kirishni ta'minlash uchun kerakli protokollarni yuritadi va tarmoq vositalarini boshqaradi.

3. Foydalanuvchi interfeysi: Bu qismat, foydalanuvchi va tizim orasida aloqani ta'minlash uchun kerakli interfeyslar va dasturlarni ta'minlaydi.

4. Dasturlar va xizmatlar: Bu qismat, foydalanuvchilar uchun kerakli dasturlarni va xizmatlarni ta'minlaydi, masalan, fayl boshqarish, tizim sozlamalari va boshqa funksiyalar.

OS funksiyalari quyidagilardir:

1. Resurslar boshqarish - OS, kompyuter resurslarini, masalan, RAM va disk yoki tarmoq resurslarini boshqaradi.

2. Dasturlarni boshqarish - OS, dasturlarni ishga tushirish, bajarish va to'xtatish imkoniyatlarini ta'minlaydi.

3. Foydalanuvchi interfeysi - OS, foydalanuvchi va tizim orasida aloqani ta'minlash uchun kerakli interfeyslar va dasturlarni ta'minlaydi.

4. Aloqa va tarmoq tuzish - OS, tarmoqqa kirishni ta'minlash uchun kerakli protokollarni yuritadi va tarmoq vositalarini boshqaradi.

5. Xavfsizlik va himoya - OS, tizimdagi ma'lumotlarni xavfsizligini ta'minlash, zararli dasturlar va viruslarga qarshi himoya ko'rsatish va boshqa xavfsizlik masalalarni hal qilish uchun ko'rsatmalarni ta'minlaydi.

OS xususiyatlari quyidagilardir:

1. Tarmoq tuzish: OS, tarmoqqa kirishni ta'minlash uchun kerakli protokollarni yuritadi va tarmoq vositalarini boshqaradi.

2. Ko'p yadroli: OS, bitta yadro yoki ko'p yadroli bo'lishi mumkin.

3. Platformaga moslashtirilishi: OS, kerakli platformaga moslashtirilgan bo'lishi kerak.

4. Ma'lumotlarni xavfsizligi: OS, tizimdagi ma'lumotlarni xavfsizligini ta'minlash, zararli dasturlar va viruslarga qarshi himoya ko'rsatish va boshqa xavfsizlik masalalarni hal qilish uchun ko'rsatmalarni ta'minlaydi.

5. Foydalanuvchi interfeysi: OS, foydalanuvchi va tizim orasida aloqni ta'minlash uchun kerakli interfeyslar va dasturlarni ta'minlaydi va foydalanuvchilar bilan o'zaro hamkorlik qilishga imkon beradi.

6. Resurslarni boshqarish: OS, kompyuter resurslarini, masalan, RAM va disk yoki tarmoq resurslarini boshqaradi va foydalanuvchilar uchun kerakli resurslarni taqsimlash uchun imkon beradi.

7. Dasturlarni boshqarish: OS, dasturlarni ishga tushirish, bajarish va to'xtatish imkoniyatlarini ta'minlaydi va dasturlar o'rtasida o'zaro hamkorlik qilishga imkon beradi.

8. O'zidan foydalanuvchi xususiyatlarini ta'kidlash: OS, foydalanuvchining xususiyatlari, masalan, til, ovozli qo'llash, kursor va boshqa xususiyatlarini tanlashga imkon beradi.

9. Qulaylik: OS, foydalanuvchining ishini osonlashtirish uchun ko'rsatmalarni ta'minlaydi va foydalanuvchining ishini tezlashtiradi.

10. Kompyuter tizimining barcha qismlari bilan integratsiya: OS, kompyuter tizimining barcha qismlari bilan integratsiyaga ega bo'lishi kerak, masalan, printer, skaner, kamera va boshqa qurilmalar.

Bu kabi xususiyatlar va funksiyalar OS-lar yordamida foydalanuvchilar va dasturchilar uchun kompyuterni oson va qulay foydalanishga imkon beradi.

Laziz Jamoliddinov, [21.06.2023 12:21]

Jarayonlar va ularni boshqarish:

Jarayon, kompyuter dasturlari yoki operatsion tizimlarida bajariladigan har qanday amalga aytishdir. Bu amallar, fayllarni yaratish, o'qish, yozish, ko'chirish, o'chirish, tashqi qurilmalar bilan aloqani ta'minlash va boshqa amallarni o'z ichiga oladi. Jarayonlar, kompyuter dasturlarida o'zaro aloqani ta'minlash, tizim resurslarini boshqarish, tarmoq bilan aloqani ta'minlash va boshqa funksiyalarni bajarish uchun kerakli vositalarni ta'minlash uchun yaratilgan tizimdir.

Jarayonlar boshqarish quyidagi usullardan foydalanadi:

1. Bajarilayotgan dasturni to'xtatish va qayta boshlash

2. Bajarilayotgan dasturning bo'sh ishga tushirilgan joylarini tekshirish

3. Operatsion tizim yordamida jarayonlarni boshqarish

4. Task manager yordamida jarayonlarni boshqarish

Xotira va ularni boshqarish:

Xotira, tizimning ma'lumotlar omborini tashkil etadi va fayllarni saqlash uchun kerakli resurslarni ta'minlaydi. Xotira, kompyuterda yuklanadigan ma'lumotlar uchun kerakli bo'sh joylarni ta'minlash uchun keraklidir. Xotira, kompyuter tizimida ishlovchi dasturlar va operatsion tizimlar uchun zaruriy bo'lgan ma'lumotlar uchun keraklidir.

Xotira boshqarish quyidagi usullardan foydalanadi:

1. Fayllarni saqlash uchun xotira bo'shligini tekshirish

2. Fayllarni saqlash uchun xotirani tozalash

3. Fayllarni saqlash uchun xotirani kengaytirish

4. Fayllarni saqlash uchun xotirani tarmoqqa ulashish

Fayllar va ularni boshqarish:

Fayl, kompyuter tizimida yaratilgan va saqlangan ma'lumotlarni ifodalovchi elementdir. Fayllar ko'p turlarda bo'lishi mumkin, masalan, matn fayllari, rasmlar, video fayllari, dasturlar va boshqalar. Fayllar, kompyuter operatsion tizimida ma'lumotlar omborini yaratish uchun keraklidir.

Fayllar boshqarish quyidagi usullardan foydalanadi:

1. Fayllarni yaratish, kiritish va saqlash

2. Fayllarni ochish, o'qish va yozish

3. Fayllarni ko'chirish, tahrirlash va o'chirish

4. Fayllarni qayta nomlash va tekshirish

Fayllar turi, ularni boshqarish usullari va foydalanish maqsadiga qarab o'zgartirilishi mumkin. Fayllarni boshqarish uchun, operatsion tizim yordamida, tarjimonlarga, ma'lumot omborini boshqarish dasturlariga va boshqa fayl boshqarish dasturlariga muhtoj bo'lish mumkin.

Laziz Jamoliddinov, [21.06.2023 12:21]

Foydalanuvchi interfeysi (FI) - bu kompyuter dasturlarida foydalanuvchilar bilan o'zaro aloqani ta'minlash uchun ko'rsatilgan qulaylik va imkoniyatlardan iborat tizimdir. Foydalanuvchi interfeysi, kompyuter dasturlarini foydalanuvchilar uchun oson va tushunarli qilishda muhim ahamiyatga ega.

Foydalanuvchi interfeysi turlari:

1. Graphical User Interface (GUI) - tizimning ma'lum bir qismini yoki to'liq ekranini grafiya yordamida ko'rsatadi.

2. Command-Line Interface (CLI) - foydalanuvchi interfeysi, kompyuter dasturlarini ko'rsatish uchun matn ko'rinishidagi buyruqlarni ishlatadi.

3. Natural Language Interface (NLI) - foydalanuvchi interfeysi, inson tili yordamida dasturlarni boshqarishga imkon beradi.

Foydalanuvchi interfeysi turlari va xususiyatlari:

1. GUI - ularning oson foydalanish, ma'lumotlar bilan ishlash, ma'lumotlar yorliqlari, sahifalar va tablar, qidirish, interaktivlik, menyu yordamida ma'lumotlarni boshqarish imkoniyati mavjud.

2. CLI - ularning buyruq yozish vaqti kam, barcha buyruqlar bo'yicha ma'lumotlar, ko'rsatkichlar va belgilashlar kabi xususiyatlar mavjud.

3. NLI - ularning inson tili yordamida dastur boshqarish imkoniyati, ma'lumotlarni boshqarish uchun so'zlar va jumlalar kiritish imkoniyati mavjud.

Foydalanuvchi interfeysi xususiyatlari:

1. Oson foydalanish - Foydalanuvchilar tizimni oson tushunishlari va ishlashlari kerak.

2. Sodda va tushunarli - Foydalanuvchilarning tizimni tushunish va ishlatish jarayoni qulay va tushunarli bo'lishi kerak.

3. Ko'prangli - Foydalanuvchilarning tizimning ko'p funksiyalaridan foydalanish imkoniyatini ta'minlashi kerak.

4. Ma'lumotlar yorliqlari - Foydalanuvchilarning tizimda ma'lumotlarni qidirish va topishlarini osonlashtirish kerak.

5. Interaktivlik - Foydalanuvchilarning tizim bilan hamkorlik qilishi va amallarini bajarish imkoniyatini ta'minlashi kerak.

Laziz Jamoliddinov, [21.06.2023 12:22]

Windows tizimi, Microsoft tomonidan ishlab chiqarilgan operatsion tizimdir. Bu tizim, bir nechta mukammal arxitektura va komponentlarga ega. Quyidagi qismlar Windows tizimi arxitekturasiga kiradi:

Windows tizimi arxitekturasi:

1. Win32 subsystem - Bu arxitektura, Windows tizimining eng muhim qismidir va ilgari Windows 95-da chiqarilgan. Ushbu subsystem, 32-bit dasturlarni ishga tushirish uchun yaratilgan.

2. Windows NT kernel - Bu arxitekturadagi kernel, Windows tizimining asosiy qismidir. Ushbu kernel, fayllarni boshqarish, xotira boshqarish va tarmoq aloqalarini ta'minlash uchun kerakli vositalarni ta'minlaydi.

3. User mode - Bu arxitekturadagi subsystem, tizimning foydalanuvchi interfeysiga qarab ishlaydi. Ushbu subsystem, foydalanuvchi interfeysini ishga tushirish, dasturlarni boshqarish va tizim resurslariga kirish uchun kerakli vositalarni ta'minlaydi.

4. Security subsystem - Bu arxitekturadagi subsystem, tizimning xavfsizlikni ta'minlash uchun yaratilgan. Ushbu subsystem, foydalanuvchilar va guruhlar orasidagi huquqlarni boshqarishga imkon beradi.

Windows tizimi komponentalari:

1. Kernel mode drivers - Bu komponent, tizimning kernelida ishlaydi va tizim resurslarini boshqarish uchun kerakli vositalarni ta'minlaydi.

2. User mode drivers - Bu komponent, tizimning foydalanuvchi interfeysiga qarab ishlaydi va dasturlarni boshqarish va tizim resurslariga kirish uchun kerakli vositalarni ta'minlaydi.

3. Services - Bu komponent, arka plandagi ishlarni bajarish uchun yaratilgan va iste'molchilarga qo'shimcha xizmatlar ko'rsatish imkonini beradi.

4. Libraries - Bu komponent, tizim funksiyalarini yaratishda va qo'shishda ishlatiladigan kodlar to'plamidir.

5. User interface - Bu komponent, foydalanuvchi interfeysini ishga tushirish uchun kerakli komponentlarni ta'minlaydi.

6. File system - Bu komponent, fayllarni saqlash va boshqarish uchun kerakli vositalarni ta'minlaydi.

7. Networking - Bu komponent, tarmoq bilan aloqani ta'minlash va tarmoqda ma'lumot almashish uchun kerakli vositalarni ta'minlaydi.

8. Security - Bu komponent, tizim xavfsizligini ta'minlash uchun kerakli vositalarni ta'minlaydi.

Laziz Jamoliddinov, [21.06.2023 12:23]

Virtual xotira (inglizcha: virtual memory) - bu kompyuterda xotira bo'sh emasligi sababli dasturlar va operatsion tizimlar uchun qo'shimcha xotira. Virtual xotira yordamida, operatsion tizim fayllarni xotiradan diskga yozib, diskdagi fayllarni esa xotiraga yuklaydi. Bu, xotirani kamroq ishlatishga imkon beradi va dasturlarni diskga yozish va xotiraga yuklash jarayonlarida tezlikni oshiradi.

Virtual xotira kirish (foydalanish) rejimlari:

1. Virtual xotira faollashtirilmasa ham dastur ishlashi mumkin, lekin xotirda ko'p joy talab qilishi mumkin. Bu rejimda, operatsion tizim xotirani optimallashtirib, dasturning zaruriy bo'lgan qismlarini xotiraga yuklaydi.

2. Virtual xotira faollashtirilsa, dasturning barcha qismlari xotirada bo'lishi kerak emas. Bu rejimda, dasturing faqat xuddi hozirgi qismi xotirada bo'lishi kerak.

3. Virtual xotira faollashtirilmasa, dastur ishlashi mumkin emas. Bu rejimda, xotira bo'sh emasligi sababli, dastur ishlay olmaydi.

Virtual xotira, operatsion tizimlarning muhim qismidir va kompyuterda dasturlarni ishlatishda katta ahamiyatga ega. Virtual xotira yordamida, xotira ishlatishni optimallashtirish va tezligi oshirish mumkin.

Laziz Jamoliddinov, [21.06.2023 12:24]

Tizimning sxemasi va komponentalari:

Kompyuter tizimi, bir nechta qurilmalar va tizimlar jamlanmasidan iborat bo'lib, boshqa tizimlar va qurilmalar bilan ham aloqada bo'lishi mumkin. Quyidagi komponentlar tizimning asosiy qismlari hisoblanadi:

1. Protsessor (CPU) - Bu komponent, tizimning barcha amallarini bajarish uchun kerakli hisoblash kuchini ta'minlaydi.

2. Xotira (RAM) - Bu komponent, tizimning qayta ishlash qobiliyatini ta'minlaydi va foydalanuvchilarning dasturlarni ishga tushirish uchun kerakli xotira miqdorini ta'minlaydi.

3. Qo'shimcha xotiralar (Hard Disk, SSD va boshqa) - Bu komponentlar, fayllarni saqlash va uzatish uchun kerakli xotiraning miqdorini ta'minlaydilar.

4. Tarmoqlar - Bu komponentlar, kompyuterlar o'rtasidagi ma'lumot almashuvini ta'minlaydilar.

5. Kengaytmalar - Bu komponentlar, qurilmalarga qo'shiladigan boshqa tizimlardir, masalan, printerlar, skanerlar, monitorlar, klaviaturalar va boshqa qurilmalar.

Qismsistemalar va bajaruvchi sistemalar:

Qismsistemalar va bajaruvchi sistemalar, tizimning boshqa qismlari hisoblanadi va ularning bir-biriga bog'liq emasligi mumkin. Quyidagi qismlar tizimning qismsistemalari va bajaruvchi sistemalari hisoblanadi:

1. Qismsistemalar - Bu qismlar, operatsion tizimni bajarish uchun kerakli foydalanuvchi interfeysini, xotira boshqarish tizimlarini, joriy bo'lish tizimlarini, tarmoq tizimlarini va boshqa qurilmalarni ta'minlaydilar.

2. Bajaruvchi sistemalar - Bu qismlar, tizimning bajarilish jarayonini ta'minlaydilar. Bajaruvchi sistemalar, operatsion tizimni boshlash, foydalanuvchi interfeysini ishga tushirish, dasturlarni boshqarish va boshqa qo'shimcha amallarni bajarish uchun kerakli xizmatlar va protsesslarni ta'minlaydilar.

Bajaruvchi sistemalar va qismsistemalar, tizimning ishlashida katta ahamiyatga ega bo'lib, tizimning barcha qismlari hamkorlikda ishlaydi. Bajaruvchi sistemalar barcha tizimni bajarish uchun kerakli xizmatlarni va tizimning qayta ishlash qobiliyatini ta'minlaydi, qismsistemalar esa barcha qurilmalarni birlashtirib operatsion tizimning to'liq ishlashini ta'minlaydi.

Laziz Jamoliddinov, [21.06.2023 12:25]

Windows muhitida programmalash asoslari:

1. Microsoft Visual Studio - Bu dasturlash muhiti, Windows tizimida dasturlarni yaratish uchun ko'p yo'l taqdim etadi. Visual Studio, C++, C#, Visual Basic va boshqa dasturlash tillarini qo'llashga imkon beradi.

2. .NET Framework - Bu Microsoft tomonidan yaratilgan dasturlash muhiti, Windows tizimida dasturlarni yaratish uchun ishlatiladi. Bu muhit, C++, C# va Visual Basic dasturlash tillariga ega.

3. Windows API - Bu dasturlash interfeysi, Windows tizimida dasturlarni yaratish uchun yordam beradi. Bu interfeys orqali, dasturchilar Windows tizimining qayta ishlash qobiliyatlariga kirish, fayllarni boshqarish, tarmoq bilan aloqani ta'minlash va boshqa amallarni bajarishga imkon beradi.

Windowsda tayanch (bazali) kontseptsyalar:

1. Registry - Bu tizim, Windowsda o'rnatilgan dasturlarning o'zgartirishlari, o'rnatilgan qurilmalar va tizim sozlamalari haqida ma'lumotlarni saqlash uchun ishlatiladi.

2. DLL - Bu tizim, ko'p qo'llaniladigan kodning bir nechta dasturlar tomonidan qo'llanilishini ta'minlaydi. DLL-fayllari, dasturlarning o'lchami kichik bo'lishiga qaramay, o'zgartirishlarni saqlash uchun yordam beradi.

3. Windows Service - Bu tizim, Windows tizimida arka plandagi ishlarni bajarish uchun ishlatiladi. Ushbu xizmatlar, tizimni avtomatik ravishda boshlashga, fayllarni avtomatik ravishda ishlatishga imkon beradi.

API va uni amalga oshirish variantlari:

Windows API, C++ dasturlash tilida ishlatiladi, lekin boshqa dasturlash tillari ham Windows API-ni qo'llab-quvvatlashadi. API-ni amalga oshirish uchun bir nechta variantlar mavjud:

1. Native API - Bu variant, C++ dasturlash tilida ishlatiladi va Windows tizimining qayta ishlash qobiliyatlariga kirish imkonini beradi.

2. Managed API - Bu variant, C# va Visual Basic dasturlash tillarida ishlatiladi va .NET Framework yordamida API-ni amalga oshirishga imkon beradi.

3. Web API - Bu variant, tarmoqdagi ma'lumot almashuvini ta'minlash uchun ishlatiladi. Web API, tarmoq protokollari yordamida ma'lumot almashuvini ta'minlaydi.

Windows API-ni amalga oshirish variantlari, dasturchilar tomonidan ishlatiladigan dasturlash tillariga bog'liq bo'lib, o'zaro aloqada bo'lishi mumkin. Bu variantlar, Windows tizimida dasturlarni yaratish va ishlatishning qulayligini oshirishga yordam beradi.

Laziz Jamoliddinov, [21.06.2023 12:25]

Dinamik komponovka qilinadigan (yuklanadigan) DLL kutubxonalar:

DLL (Dynamic Link Library) kutubxonalar, Windows tizimida yuklanadigan va ishlatiladigan dasturlarni qo'llab-quvvatlash uchun ishlatiladi. Bu kutubxonalar, dastur ishlatilmayotgan vaqtida yuklanadi va dastur tomonidan talab qilinganida ishga tushiriladi. DLL kutubxonalar, dasturlarni kichiklashtirishga, ishga tushurish tezligini oshirishga va kodni qayta ishlatishga imkon beradi.

Funksiyalarni boshlangʻich qiymatlarini o‘rnatish:

Funksiyalarni boshlangʻich qiymatlarini o'rnatasiz, ularning ishlatishini ta'minlay olmaysiz. Boshlang'ich qiymatlar, funksiyalar tomonidan qabul qilingan ma'lumotlarni to'g'ridan-to'g'ri qaytaradi.

Funksiyalarni boshlangʻich qiymatlarini o‘rnatish, eksport va import qilish:

Funksiyalarni DLL kutubxonalarida o'rnatish uchun export va import qilish kerak. Export qilish, DLL kutubxonasidagi funksiyalarni boshqa dasturlarga yuklash imkonini beradi. Import qilish, DLL kutubxonalaridan funksiyalarni chaqirish imkonini beradi.

Dasturlash tillari va DLL kutubxonalarni ishlab chiqish:

DLL kutubxonalarini ishlab chiqish uchun, dasturchilar C++ dasturlash tilidan foydalanishadi. Bu til, DLL kutubxonalarini yaratish, funksiyalarni o'rnatish va export qilish uchun yordam beradi. DLL kutubxonalarini yaratish, funksiyalarini o'rnatish va import qilish uchun Microsoft Visual Studio dasturlash muhiti yoki boshqa dasturlash dasturlari ishlatiladi.

Laziz Jamoliddinov, [21.06.2023 12:26]

Programmalash tizimlari:

Programmalash tizimlari, dasturlarni yaratish va ishga tushirish uchun ishlatiladigan yordamchi dasturlash vositalari. Bu tizimlar, kodlarni yozish, tekshirish, yuklash va boshqa amallarni bajarish uchun kerakli vositalarni ta'minlaydi.

Translyatsilash texnikasi:

Translyatsilash texnikasi, dastur kodi tili bilan yozilgan kodni kompyuter tili bilan o'girish uchun ishlatiladi. Bu texnika, dastur kodi tili bilan yozilgan kodni kompyuter tili bilan o'girishga yordam beradi.

Programmalash tizimlari va translyatsilash texnikasi:

Programmalash tizimlari va translyatsilash texnikasi, dasturlarning yaratilishi va ishga tushirilishi uchun kerakli vositalarni ta'minlashda yordam beradi. Dastur kodi tili bilan yozilgan kodni translyatsiya qilish, kompyuter tili bilan o'girish uchun kerakli bo'lgan dasturlash tizimlari yordam beradi.

Dasturlash muhiti va uning komponentalari:

Dasturlash muhiti, dasturni yaratish uchun kerakli barcha vositalar va resurslarni o'z ichiga olgan dasturlash muhitidir. Bu muhit, dastur yaratishida, kodlarni yozishda, tekshirishda, kompyuter tili bilan o'girishda, dasturlarni boshqarishda va boshqa amallarda yordam beradi. Dasturlash muhiti, yozilgan kodni kompyuter tili bilan o'girish uchun kompilyator, yuklash uchun linker, dasturlarni boshqarish uchun interfeys va boshqa komponentlarga ega bo'lishi mumkin.

Dasturlash tili (DT) strukturasi:

Dasturlash tili strukturasi, dasturlash tili yozilgan kodning tuzilishi va shaklini ma'lum qiluvchi muhim elementlardan biridir. DT-strukturasi, dasturlash tilida o'zgaruvchi, massiv, funksiya, klass, interfeys va boshqa elementlarning tuzilishini aniqlaydi. DT-strukturasi, dasturchilarga yozilgan kodni tushunarli va qulay qilishda yordam beradi. DT-strukturasi, C++, Java, Python va boshqa ko'plab dasturlash tillarida ishlatiladi.

Laziz Jamoliddinov, [21.06.2023 12:27]

Translyatorlar turi:

Translyatorlar, dastur kodi tili bilan yozilgan kodni boshqa tilda o'girish uchun ishlatiladigan vositalardir. Bu translyatorlar turli xil bo'lishi mumkin, masalan, kompyuter tili bilan yozilgan kodni inson tili bilan o'girish uchun tarjimonlar yoki dastur kodi tili bilan yozilgan kodni boshqa dasturlash tillariga o'girish uchun kompilyatorlar mavjud.

Translyatsiyalash sxemasi:

Translyatsiyalash sxemasi, dastur kodi tili bilan yozilgan kodni boshqa tilda o'girish jarayonini ifodalovchi sxema hisoblanadi. Bu sxema, dastur kodi tili bilan yozilgan kodning tahlilini, tuzilishini, o'girishini va natijaviy kodni yaratishni ta'minlaydi.

Dasturlash tili (DT) strukturasi va translyatorlar turi:

Dasturlash tili strukturasi, dastur kodi tili bilan yozilgan kodning tuzilishi va shaklini aniqlaydi. Bu struktura, translyatorlar uchun muhim elementlar taqdim etadi, shuningdek, translyatsiyalash sxemalarini tuzishda ham yordam beradi. Translyatorlar, DT-strukturani o'qib chiqish, kiritish, yozish va boshqa amallarni bajarish uchun ishlatiladi.

Dasturlash tili (DT) strukturasi va translyatsiyalash sxemasi:

DT-struktura va translyatsiyalash sxemasi, dastur kodi tili bilan yozilgan kodni boshqa tilda o'girishning muhim qismlari hisoblanadi. DT-struktura, kod tuzilishini va shaklini aniqlash uchun yordam beradi, translyatsiyalash sxemasi esa kodni o'girish jarayonini tushuntiradi. DT-strukturani tahlil qilish, translyatsiyalash sxemasi tuzish va natijaviy kodni yaratish uchun translyatorlar ishlatiladi.

Dasturlash tili (DT) strukturasi, translyatorlar turi va translyatsiyalash sxemasi:

DT-struktura, translyatorlar va translyatsiyalash sxemasi, dastur kodi tili bilan yozilgan kodni boshqa tilda o'girish jarayonini tushuntiradigan muhim elementlar hisoblanadi. DT-strukturasi, kod tuzilishini aniqlashda yordam beradi, translyatorlar turli xil translyatsiya jarayonlarini bajarish uchun ishlatiladi, va translyatsiyalash sxemasi kodni o'girishning muhim qismlarini ifodalaydi. Bu elementlar birgalikda, dastur kodi tili bilan yozilgan kodni boshqa tilda o'girish uchun kerakli muhitni ta'minlaydilar.

Laziz Jamoliddinov, [21.06.2023 12:28]

Kutubxonalarni yaratish:

Kutubxonalar, dasturlarni yaratishda qo'llanadigan katta kod to'plamlaridir. Bu kodlar, bir yoki bir nechta funksiyalarni o'z ichiga oladi va boshqa dasturlarning foydalanishi uchun yuklanadi. Kutubxonalar, dastur kodini kichiklashtirish, ishga tushirish tezligini oshirish va muammolarni hal qilishda yordam beradi. Kutubxonalar C++, Java, Python va boshqa dasturlash tillarida yaratilishi mumkin.

Subjarayonlarni tashkil qilish:

Subjarayonlar, kutubxonalar yoki dasturlardagi kodni bitta joyga to'plamash uchun ishlatiladigan usul hisoblanadi. Bu usul yordamida kod bloklari tashkil qilinadi va ularga nomlar beriladi. Subjarayonlar, kodni tashkil qilishda va qayta ishlatishda qulaylik yaratadi.

Konsollar va dasturlarni ishlab chiqish texnikasi (bosqichlari):

Konsollar, dastur yaratishda va ishlatishda foydalaniladigan grafikali interfeyssiz dasturlash vositalardir. Konsollar yordamida foydalanuvchi bilan o'zaro muloqot o'rnatiladi va dastur natijalari konsol ekranida ko'rsatiladi. Dasturlarni ishlab chiqish bosqichlari esa kodni yozishdan boshlab, tekshirishdan o'tkaziladi va keyingi bosqichlarda funksiyalarni tuzish, test qilish va kutubxonalar bilan birgalikda ishlatishni o'z ichiga oladi.

Dasturlashning klassik (o‘rnakli, namunali) metodlari:

Dasturlashning klassik yoki o'rnakli metodlari, dasturlarni yaratishda qo'llanadigan usullardan biridir. Bu metod yordamida dastur tuzish bosqichlari, kodni yozish, tekshirish, funksiyalarni yaratish, test qilish va chiqishga tayyorlash bosqichlari kabi muhim qadamlar aniqlanadi.

Modullar turi:

Modullar, dasturlarni tuzishda qo'llanadigan kod bloklaridir. Bu bloklar, bir yoki bir nechta funksiyalarni o'z ichiga oladi va boshqa dasturlarning foydalanishi uchun yuklanadi. Modullar, kutubxonalar bilan bir xil bo'lib, dasturlarni kichiklashtirish, ishga tushirish tezligini oshirish va muammolarni hal qilishda yordam beradi. Modullar C++, Java, Python va boshqa dasturlash tillarida yaratilishi mumkin.

Laziz Jamoliddinov, [21.06.2023 12:29]

Tahlil bosqichlari:

Dastur kodini tahlil qilishda qo'llanadigan bosqichlar, kodni tuzilishi, sintaksisi va xatoliklarni aniqlash uchun foydalaniladi. Tahlil bosqichlari quyidagilardan iborat bo'lishi mumkin:

1. Leksik tahlil: Kodni to'g'ridan-to'g'ri harflarga ajratish bosqichi. Bu bosqichda, koddagi harflar va belgilar, sonlar, operatorlar va boshqa belgilar belgilanadi.

2. Sintaksis tahlili: Kodning sintaksisini tekshirish bosqichi. Bu bosqichda, koddagi sintaksis xatoliklari, masalan, bitta operatorni ikki marta ishlatish kabi muammolarni tekshirish uchun foydalaniladi.

3. Ma'noga tahlil: Kodning to'g'ri ishlashi uchun ma'noga tekshirish bosqichi. Bu bosqichda, kodning kelgusi natijalarini tekshirish uchun testlar yaratiladi va kod natijalari tekshiriladi.

Optimallashtirish usullari:

Dastur kodining optimallashtirishida qo'llanadigan usullar, kodni ishlash tezligini oshirish, xotiradagi joyni kamaytirish va boshqa muammolarni hal qilishda yordam beradi. Quyidagi optimallashtirish usullari bilan foydalaniladi:

1. Algoritmlarni optimallashtirish: Muammolarni yechishda qo'llanadigan algoritmlarni yangi usullarga asoslash va shunchaki ishlatish orqali, kodni tezlashtirish uchun foydalaniladi.

2. Kode optimallashtirish: Kodni tahlil qilish va boshqa optimallashtirish usullari yordamida, kod yozish paytida qo'shimcha xotira kerak bo'lgan joylarni identifikatsiya qilish, kodning ishlash tezligini oshirish va xotira ishlatishni kamaytirish uchun o'zgartirishlar kiritish orqali, kodni optimallashtirish uchun foydalaniladi.

3. Kompilyatsiya va taraqqiyotli optimallashtirish: Kodni tahlil qilish, boshqa optimallashtirish usullari, JIT-kompilyatsiya va boshqa yuqori darajadagi optimallashtirish texnikalari yordamida, kodni optimallashtirish uchun foydalaniladi. Bu usullar kodni tezlashtirish uchun yordam beradi va dastur ishga tushirilgandan so'ng yuqori tezlikda ishlaydi.

Laziz Jamoliddinov, [21.06.2023 12:31]

Kompyuter dasturiy ta'minoti:

Kompyuter dasturiy ta'minoti, dasturlar va tizimlar yaratish uchun muhim bo'lgan vositalar to'plamidir. Bu ta'minot quyidagi komponentlardan iborat bo'lishi mumkin:

1. Dasturlash tillari: Dasturlash tillari, dasturlar yaratishda foydalaniladigan kod yozish tillaridir. Masalan, Python, Java, C++, C# va boshqalar.

2. Qo'llanma kitoblari: Qo'llanma kitoblari, dasturlarni yaratish, ishlatish va yechishda yordam beruvchi dokumentatsiyalardir.

3. Integrlangan o'rnatish yordamlari: Bu yordamlar, dasturlarni tizimga o'rnatish va boshqa vositalarni yuklash uchun foydalaniladi.

4. Tizim muharriri: Tizim muharriri, dasturlarni yaratishda va ishlatishda foydalaniladigan komponentlardan biridir. Bu muharrir, kod yozish, tahrir qilish va tekshirish kabi vazifalarni bajarish uchun qulaylik yaratadi.

Kompyuter arxitekturasi:

Kompyuter arxitekturasi, kompyuter tizimining qurilishini va ishga tushirishini ta'minlaydigan asosiy qurilmalarni o'z ichiga oladi. Bu arxitektura quyidagi qurilmalardan iborat bo'lishi mumkin:

1. Mikroprotsessorlar: Mikroprotsessorlar, kompyuter tizimining asosiy ishlab chiqaruvchi bo'limi hisoblanadi. Bu bo'lim, kompyuterda amalga oshiriladigan barcha hisob-kitoblarni bajaradi.

2. Xotira: Xotira, kompyuter tizimining ma'lumotlarini saqlash uchun qo'llanuvchi joy hisoblanadi. Bu tizim, ma'lumotlar saqlash va ko'chirish, dasturlarni ishga tushirish, va boshqa vazifalarni bajarishda ishlatiladi.

3. Tizim busi: Tizim busi, kompyuter tizimining barcha qurilmalarining bir-biriga bog'lanishini ta'minlaydigan tizim hisoblanadi.

SHK platformasi:

SHK platformasi, dasturlash tili yoki dasturlash tizimining ishlaganida foydalaniladigan asosiy qurilmalarni o'z ichiga oladi. SHK platformasi quyidagi qurilmalardan iborat bo'lishi mumkin:

1. Dasturlash tili: SHK platformasi, qo'llanadigan dasturlash tilini belgilaydi. Bu til, dasturlar yaratishda va ishlatishda foydalaniladi.

2. Dasturlash tizimi: SHK platformasi, dasturlar yaratish uchun foydalaniladigan dasturlash tizimini o'z ichiga oladi. Bu tizim, dasturlarni tuzish, yuklash va ishlatishda foydalaniladi.

SHK konfiguratsiyasi:

SHK konfiguratsiyasi, SHK platformasining ishga tushirilishi uchun muhim bo'lgan qurilmalar va sozlashlardan iboratdir. SHK konfiguratsiyasi quyidagi qurilmalardan iborat bo'lishi mumkin:

1. SHK platformasining versiyasi: Bu versiya, SHK platformasining belgilangan versiyasini aniqlaydi.

2. Qo'llanuvchi sozlashlari: Qo'llanuvchi sozlashlari, dastur ishga tushirilganda foydalanuvchi tomonidan ko'rsatilgan sozlashlarni o'z ichiga oladi. Masalan, til, qurilma turi va boshqa muhim sozlashlar.

SHK asosiy platasi va qurilmalari:

SHK asosiy platasi, SHK platformasini ishga tushirish uchun muhim bo'lgan qurilmalardan biridir. SHK asosiy platasi quyidagi qurilmalardan iborat bo'lishi mumkin:

1. Mikroprotsessor: Mikroprotsessor, SHK platformasining asosiy qurilmasidir. Bu qurilma, dasturlar yaratish, ishlatish va saqlashda foydalaniladi.

2. Xotira: Xotira, kompyuter tizimining ma'lumotlarini saqlash uchun qo'llanuvchi joy hisoblanadi. Bu tizim, ma'lumotlar saqlash va ko'chirish, dasturlarni ishga tushirish, va boshqa vazifalarni bajarishda ishlatiladi.

3. Tizim busi: Tizim busi, SHK platformasining barcha qurilmalarining bir-biriga bog'lanishini ta'minlaygan tizim hisoblanadi.

SHK qurilmalari:

SHK qurilmalari, SHK platformasini ishga tushirish uchun foydalaniladigan qurilmalardir. SHK qurilmalari quyidagi qurilmalardan iborat bo'lishi mumkin:

1. Grafik adapteri: Grafik adapteri, tizimning rasmlar va videolarni ko'rsatish uchun foydalaniladi.

2. Qo'llanma kitoblari: Qo'llanma kitoblari, dasturlarni yaratish, ishlatish va yechishda yordam beruvchi dokumentatsiyalardir.

3. Klaviatura va musiqa qurilmalari: Bu qurilmalar, tizimga ma'lumot kiritish va tizimni boshqarishda foydalaniladi.

4. Internet qurilmalari: Bu qurilmalar, internetga ulanish va internetdan ma'lumot olishda foydalaniladi.

Laziz Jamoliddinov, [21.06.2023 12:32]

Dasturlashning klassik metodlari:

1. Strukturalash (tuzilishga ajratish). Dasturni kichik, o'zaro bog'liq tuzilmalarga ajratish. Masalan, dastur tarkibida ma'lumotlar, konstantlar, funksiyalar, sinflar tuzilmalari bo'lishi mumkin.

2. Abstraktlashtirish. Umumiy holdagi tushunchalarni ajratib olish. Masalan, Hayvon sinfiga quyon va mushuk kabi maxsus hayvon turlari uchun sinflar yaratish.

3. Inkapsulatsiya. Ma'lumotlarni va metodlarni birgalikda saqlash. Masalan, shaxs haqidagi ma'lumotlar (isim, familiya, yosh) va shu ma'lumotlar bilan ishlov beradigan metodlarni (getIsim(), getYosh() va hokazo) Shaxs sinfida birgalikda saqlash.

4. Nasledovaniye (meros olish). Umumiy belgilarga ega bo'lgan sinflardan iborat ierarxiya yaratish. Masalan, Shaxs > Talaba > Magistrant ierarxiyasi.

5. Polimorfizm. Bir interface yoki superclass dan meros olingan ko'p sonli sinflarda bir xil nomdagi metodlarni qo'llash. Masalan, ovozBer() metodini Hayvon, Qush, Sichqon sinflarida qo'llash.

Modullar turi:

- Kutubxona modullari - boshqa modullarga mos funksionallik va ma'lumotlar yetkazib beruvchi modullar.

- Dasturiy modul - muayyan vazifani bajaradigan kodni o'z ichiga oluvchi modul.

- Test modullari - dasturiy modullarni tekshirish uchun mo'ljallangan modullar.

😎😎😎 OMAD HAMMAGA!!!

**😎 LDEV 😎**