

OBYEKT MUVOZANATINI ROSTLASHDA DASTURIY VOSITA ARXITEKTURASINING SINFLARI VA FUNKSIONAL VAZIFALARI

Buriyev Bobur Erkinovich

Alfraganus universiteti Raqamli texnologiyalar kafedrasi o'qituvchisi

<https://doi.org/10.5281/zenodo.14545596>

Annotatsiya. Ushbu tezisdagi obyekt muvozanatini rostlashda dasturiy vosita arxitekturasining sinflari va funksional vazifalari to'g'risida fikr yuritilgan. Unda dasturiy vosita arxitekturasini ishlab chiqishning asosiy vazifalari, sinflarga ajratilishidagi ahamiyati, tamoyillari va bir qancha xususiyatlari borasida ma'lumotlar berilgan.

Kalit so'zlar: interfeys, konveyer, dasturiy vosita, xabar porti, obyekt muvozzanti.

Abstract. This thesis discusses the classes and functional tasks of software architecture in object balancing. It provides information on the main tasks of software architecture development, the importance of its division into classes, principles and some of its features.

Key words: interface, pipeline, software, message port, object balance.

Dasturiy vosita arxitekturasini – bu dasturiy vositani o'zaro ish ko'ruvchi ichki bo'limlarning qandaydir bir to'plami sifatida ifodalashdir. Bunday ichki tizimlar sifatida odatda alohida dasturlar ish ko'radi. Arxitekturaning ishlab chiqish dasturiy vositaning murakkabligi bilan bo'ladigan kurashning birinchi bosqichidir, unda bog'liqmas tashkil etuvchilarga nisbatan ajratish tamoyili amalga oshiriladi.

Dasturiy vosita arxitekturasini ishlab chiqishning asosiy vazifalari sifatida quyidagilarni keltirish mumkin:

- dasturiy ichki tizimlarni ajratish va ularda dasturiy vositaning tashqi (tashqi bayonda berilgan) funksiyalarini ifodalash;
- ajratilgan dasturiy ichki tizimlar o'rtasida o'zaro ish ko'rish usullarini aniqlash.

Bu bosqichda qabul qilinadigan yechimlar hisobga olinib, keyingi funksional spetsifikatsiyaning aniqlanishi amalga oshiriladi.

Dasturiy vosita arxitekturasining asosiy sinflari sifatida quyidagi dasturlar asosiy mezon sanaladi:

- yaxlit dastur;
- avtonom bajariladigan dasturlar kompleksi;
- qatlamli dasturiy tizim;
- parallel bajariladigan dasturlar jamoasi.

Yaxlit dastur dasturiy vosita arxitekturasining ifodali hodisasini ifodalaydi: Dasturiy vosita tarkibiga faqat bitta dastur kiradi. Bunday arxitektura odatda, dasturiy vosita qandaydir bir aniq yorqin ifodalangan funktsiyani bajarishi kerak bo'lgan va uning amalga oshirilishi juda murakkab bo'lmagan holda tanlanadi. Avtonom bajariladigan dasturlar kompleksi shunday dasturlar majmuidan iborat bo'ladiki:

- bu dasturlardan ixtiyoriy bittasi foydalanuvchi tomonidan faollashtirilishi (ishga tushirilishi) mumkin;
- faollashtirilgan dasturni bajarishda shu majmuaning boshqa dasturlari, faollashtirilgan dastur o'z ishini tugatmagunicha, faollashtirilishi mumkin emas;
- bu majmuaning hamma dasturlari bitta axborot muhitida qo'llaniladi.

Shunday qilib, bu majmuaning dasturlari boshqarishi bo'yicha o'zaro ish ko'rishmaydi, ular o'rtasida o'zaro ish ko'rish faqat umumiy axborot muhiti orqali amalga oshiriladi. Qatlamli dasturiy tizim qatlamlar deb ataluvchi, tartibga solingan dasturlar ichki tizimining qandaydir bir to'plamidan iborat, bunda:

har bir qatlamda keyin keladigan (ancha yuqori) qatlamlarning xossalari (va mavjud bo'lishi) haqida hech narsa ma'lum emas;

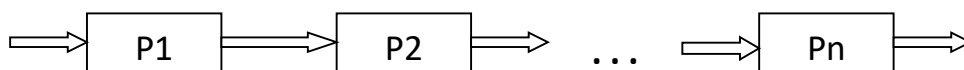
har bir qatlam boshqarish (komponentga murojaat qilish) bo'yicha bevosita undan oldin kelgan (ancha past) qatlam bilan, oldindan aniqlangan interfeys orqali, oldin keladigan hamma qatlamlarning ichki tuzilishi haqida hech narsa bilmasdan, o'zaro ish ko'rishi mumkin;

Har bir qatlam ma'lum resurslarga ega, qatlam uni boshqa qatlamlardan berkitadi, yoki ularning ba'zi bir mavhumlarini bevosta keyingi qatlamga (ko'rsatilgan interfeys orqali) taklif etadi.

Bundan tashqari, dasturiy tizimda har bir qatlam ma'lumotlarning qandaydir mavhumligini amalga oshirishi mumkin. Qatlamlar o'rtasidagi aloqalar har bir qatlam murojaat parametrlari qiymatlarini pastdan aralash bo'lgan qatlamga uzatish va bu murojaat natijasini quyi qatlamdan yuqorisiga uzatish bilan cheklangan. Global ma'lumotlarni bir necha qatlamlar bilan ishlatish mumkin emas.

Parallel ish ko'ruvchi dasturlar jamoasi bajarilish bosqichida bo'la turib, bir vaqtda o'zaro ish ko'rishga qobiliyatli dasturlar majmuasini ifodalaydi. Bunday dasturlar, birinchidan, tezkor xotiraga chaqirilganligini, faollashtirilganligini va navbat bilan vaqt bo'yicha bitta yoki bir nechta markaziy protsessorlarni bo'lishib olishi, ikkinchidan esa, bazasida sinxronizatsiyasi hosil qilinadigan o'zaro dinamik (bajarilish jarayonida) ish ko'rishlarni amalga oshirish mumkinligini bildiradi. Odatda bunday jarayonlar o'rtasida o'zaro ish ko'rishlar bir-biriga qandaydir bir xabarni uzatish yo'li bilan amalga oshiriladi.

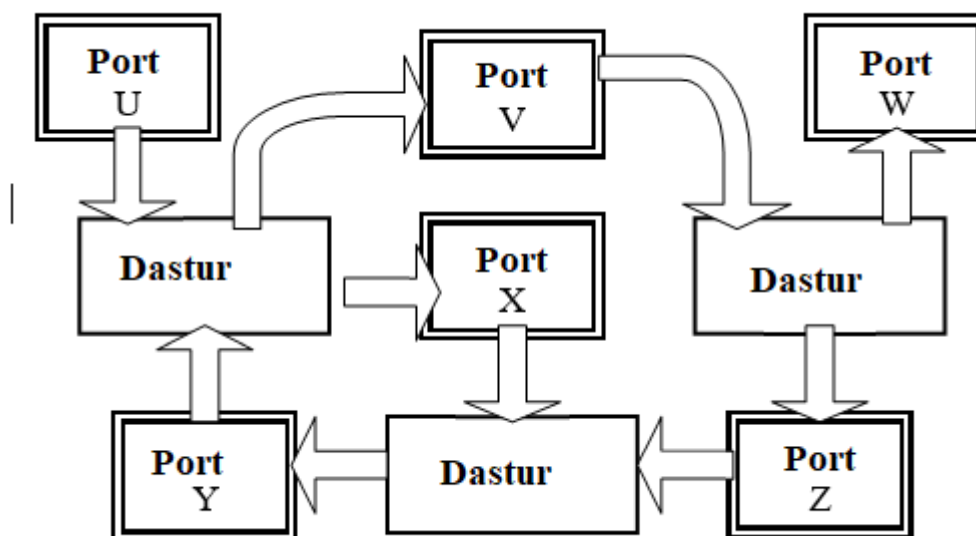
Bunday arxitekturaning eng oddiy ko'rinishi konveyerdir. Konveyerni tashkil qilish imkoniyatlari mavjud, masalan, UNIH operatsion tizimida. **Konveyer** dasturlar ketma-ketligini ifodalaydi, unda har bir dasturning standart chiqishi, eng oxirgisidan tashqari, shu ketma-ketlikning navbatdagi dasturining standart kirishi bilan bog'liq bo'ladi (1-rasm):



1-rasm. Parallel ishlovchi dasturlar konveyeri

Umumiy holda parallel ish ko'ruvchi dasturlar jamoasi tizimga xabar portlari bilan tashkil qilinishi mumkin. *Xabar porti* xabarlarning qandaydir bir navbatiga xizmat qiluvchi dasturiy ichki tizimni ifodalaydi. U dasturdan saqlash uchun, qandaydir bir xabarni, uni navbatga qo'yib turib, qabul qilishi va navbatdagi xabarni boshqa dasturga, uning talabiga ko'ra berishi mumkin. Qandaydir bir dastur tomonidan ma'lum bir portga uzatilgan xabar, endi bu dasturga tegishli bo'lmay qoladi va uning resurslarini ishlata olmaydi, ammo u, navbat tarzida boshqa biror dasturga, shu dasturning so'rovi bilan uzatilmagunicha, boshqa biror dasturga ham tegishli bo'lmaydi. Shunday qilib, xabar beruvchi dastur, bu xabarni qabul qiluvchi dastur uni qayta ishlashga tayyor bo'lmagunicha (agar qabul qiluvchi port to'lib qolmagan bo'lsa), kutish holatida bo'lmaydi.

Xabar portlariga ega dastur tizimiga misol 2-rasmda keltirilgan. U port bu rasmda keltirilgan parallel harakat qiluvchi dasturlar jamoasi uchun kiritiluvchi xabarlar porti kabi, W – esa, dasturlarning bu jamoasi uchun chiqarilayotgan xabarlar porti kabi qaralishi mumkin.



2-rasm. Xabarlar portlariga ega dasturiy tizimga misol.

Arxitektura funksiyalari. Ichki tizimlar o`rtasida o`zaro ishlarni ta`minlash uchun bir qator hollarda qandaydir bir qo`shimcha dasturiy tashkil etuvchilarni (tashqi funksiyalarni amalga oshirishdan tashqari) yaratish talab qilinmaydi – buning uchun oldindan belgilangan kelishuvlar va bazaviy dasturiy ta`minotning (operatsion tizim) standart imkoniyatlari yetarli bo`lishi mumkin. Avtonom bajariladigan dasturlar majmuasida o`zaro ishlashni ta`minlash uchun umumiy tashqi axborot muhitini va dasturni ishga tushirish uchun operatsion tizimning imkoniyatlarini bayon etish yetarli hisoblanadi. Qatlamli dasturiy tizimda ajratilgan dasturiy qatlamlarning spesifikatsiyalari va protseduralarga murojaat etishning odatdagi apparati yetarli bo`lishi mumkin. Dasturiy konveyerida dasturlar o`rtasida o`zaro ish ko`rishni shuningdek operatsion tizim (bu UNIX operatsion tizimida bo`lgani kabi) ham ta`minlab berishi mumkin.

Ammo bir qator hollarda dasturiy ichki tizimlar o`rtasida o`zaro ish ko`rishni ta`minlash uchun qo`shimcha dasturiy komponentlarni yaratish talab qilinishi mumkin. Avtonom bajariluvchi dasturlar kompleksining ishini boshqarish uchun ko`pincha maxsus buyruq interpretatori yaratiladi, u operatsion tizim ishlatadigan bazaviy buyruq interpretatoriga qaraganda talab qilinayotgan tashqi ahborot muhiti va ishga tushiriladigan dasturni tayyorlash uchun mazkur predmet sohasida ancha qulaydir.

Qatlamli dasturiy tizimda qatlam protseduralariga murojaatning alohida apparatida bu protseduralarning parallel bajarilishini ta`minlovchi tizimi yaratilishi mumkinligini aytish mumkin.

Xulosa o`rnida shuni aytish mumkinki, dasturiy ta`minot arxitekturasi sodda qilib aytganda, tizimning umumiy tuzilishi va tashkil etilishi bo`lib, dasturning turli komponentlarining bir-biri bilan o`zaro ta`sirini tavsiflaydi. Arxitektura tizimni komponentlar deb ataladigan turli qismlarga ajratadi va komponentlar, o`z navbatida, interfeyslar orqali bir-biri bilan o`zaro ta`sir qiladi.

References:

1. Шилов С.Н. Реализация инфраструктуры распределенной хеш-таблицы в рамках кластерной системы DNS / С.Н. Шилов, С.Д. Кургалин, А.А. Крыловецкий // Вестник ВГУ. Серия: Системный анализ и информационные технологии. - № 2. — Воронеж: изд-во Воронежского гос. ун-та, 2012.-С. 74-79.
2. Erkinovich, B. B. (2024). OBYEKT MUVOZANATINI ROSTLASHDA AVTOMATLASHTIRISH ALGORITMI VA DASTURIY VOSITA YARATISHNING AVTOMATIK BOSHQARISH FUNKSIYALARI. *Eurasian Journal of Mathematical Theory and Computer Sciences*, 4(1), 54-60.
3. Керниган Б. Язык программирования Си / Б. Керниган, Д. Ритчи // - 2-е изд. - М.: Вильямс, 2007. - 304 с.
4. Zaripov, Odiljon, and Shahlo Zaripova. "DINAMIK BOSHQARISH OBYEKTLAR HOLATINI BAHOLASH MASALALARINING DASTURIY TA'MINOTI." *Innovatsion texnologiyalar* 50.02 (2023): 73-77.
5. Кнут Д. Искусство программирования. Сортировка и поиск / Д. Кнут. -Т. 3. - 2-е изд. - М.: Вильямс, 2007. - 824 с.
6. Maxkamovna, M. N. (2023). SANOAT KORXONALARIDA ISHLAB CHIQRISH VA DISPETCHERLIK BOSHQARUV JARAYONLARINI (SCADA) AVTOMATLASHTIRISH. *ARXITEKTURA, MUHANDISLIK VA ZAMONAVIY TEXNOLOGIYALAR JURNALI*, 2(2), 1-3.
7. Erkinovich, B. B. AVTOMATLASHTIRISH TIZIMLARIDA OBYEKT BARQARORLIGINI NAZORAT QILISH VA MODELLASHTIRISHGA DOIR AYRIM MULOHAZALAR.