



MPI VA THREADING O'RTASIDAGI ASOSIY FARQLAR TEXNOLOGIYALAR

Ibragimova Muyassar Nazarali qizi

Ixrorva Surayyo Isroiljon qizi

Toshkent Davlat Tibbiyot Universiteti

Izoh. Umuman olganda, agar MPI parallel va Open MP parallel implementatsiyasi o'rtasida tanlov mavjud bo'lsa, ishlash jihatidan eng maqbul tanlov nima? Dasturchi bu paketning qaysi qismlari eng ko'p ishlatilishiga bog'liq deb taxmin qiladi. Shunga qaramay, biz ularning farqini batafsil tushuntirmaymiz. Ushbu mavzu MPI va Threadingning asosiy farqlariga asoslanadi,

Kalit so'zlar: *Xabar uzatish interfeysi, Open MP, Threading,*

Annotatsiya. *V obshhem, esli u vas est vybor mejdu parallelnoy realizatsiey MPI va Open MP, qanday optimalnyy vybor, proizvoditelnost? Programmist predpolagaet, chto bu zavisit ot togo, qanday paketlar naibolee chaspolzuyutsya. Tem ne mene, my ne bole ob'yasnyaem, v chem ix otlichie. Eta mavzu po suti otlichaetsya ot MPI va Threading.*

Kuluchevoy slova: *Interfeys peredachi soobshcheniy, Otkryt MP, Threading*

Annotatsiya. *Umuman olganda parallel MPI va Open MP parallel tizimlardan birini tanlashda optimal usulni tanlash qiyin masala. Dasturchilar ishlab chiqarish chiqilish qismidan yuk holda tanlovni amalga oshirish. Halichacha biz bu kutubxonalarning farqlarini to'liq muhokama qilmaymiz. Bu maqolada, MPI va Threading texnologiyasini o'xshash bo'lmagani yoritilgan.*

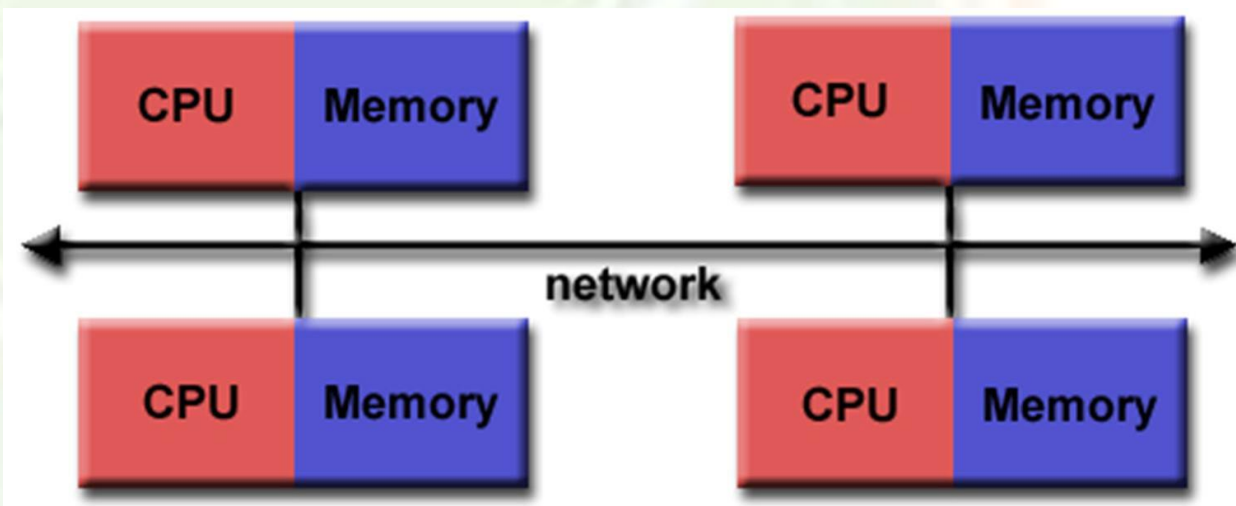
Kalit so'zlar: *Xabar yuborish interfeysi, Open MP, Threading,*

Kirish. Xabarlarni uzatish interfeysi standarti (MPI) - bu sotuvchilar, tadqiqotchilar, dasturiy ta'minot kutubxonasi ishlab chiquvchilari va foydalanuvchilar kabi 40 dan ortiq ishtirokchi tashkilotlarni o'z ichiga olgan MPI forumining konsensusiga asoslangan xabarlarni uzatish kutubxonasi standarti. Xabarlarni uzatish interfeysining maqsadi xabarlarni uzatish dasturlarini yozish

uchun keng qo'llaniladigan ko'chma, samarali va moslashuvchan standartni yaratishdir. Shunday qilib, MPI birinchi standartlashtirilgan, sotuvchidan mustaqil, xabarlarini uzatish kutubxonasidir. MPI yordamida xabarlarini uzatish dasturini ishlab chiqishning afzalliklari ko'chma, samaradorlik va moslashuvchanlik dizayn maqsadlariga juda mos keladi. MPI IEEE yoki ISO standarti emas, lekin aslida HPC platformalarida xabarlarini uzatish dasturlarini yozish uchun "sanoat standarti" ga aylandi.

Mavzular bitta dastur ichida bir vaqtning o'zida bir nechta kod yo'llarini bajarish imkonini beruvchi bir nechta texnologiyalardan biridir. Kompyuter fanida "mavzu" - bu ijro etish threadining qisqartmasi. Mavzular - bu dasturning o'zini ikki yoki undan ortiq bir vaqtning o'zida (yoki soxta bir vaqtning o'zida) ishlaydigan vazifalarga ajratish ("bo'linish" deb ataladi) usuli. Mavzular va jarayonlar bir operatsion tizimdan boshqasiga farq qiladi, lekin umuman olganda, thread jarayon ichida joylashgan va bir xil jarayondagi turli threadlar bir xil resurslarni baham ko'radi, bir xil ko'p vazifali operatsion tizimdagi turli jarayonlar esa bunday qilmaydi. Mavzular, ular iste'mol qiladigan tizim resurslari jihatidan, jarayonlarga nisbatan yengil.

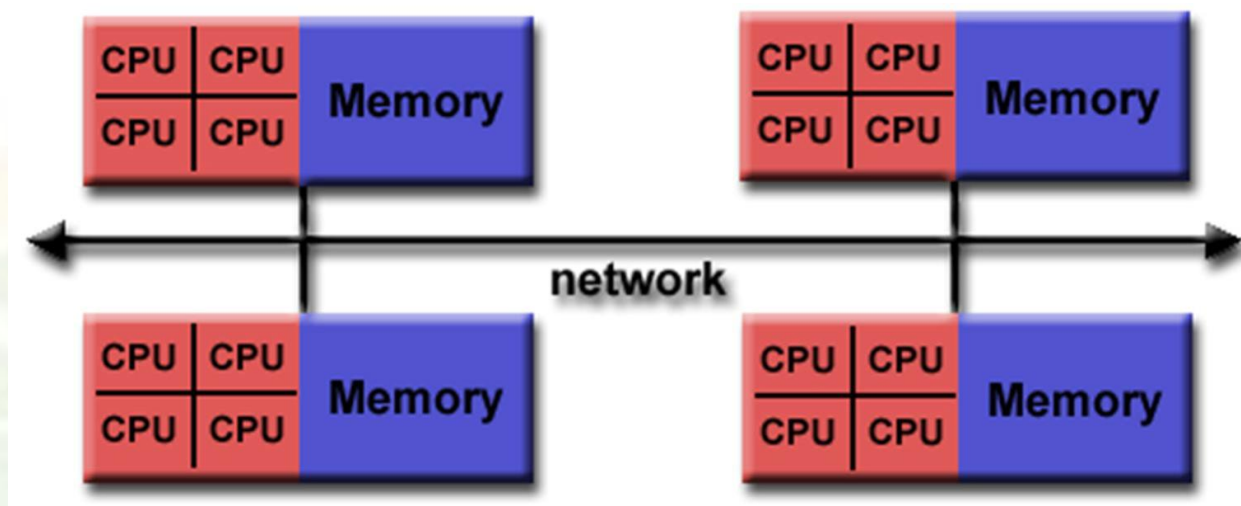
Xabarlarini uzatish interfeysi (MPI) dasturlash modeli : Dastlab , MPI o'sha paytda (1980-yillar - 1990-yillarning boshlari) tobora ommalashib borayotgan taqsimlangan xotira arxitekturalari uchun mo'ljallangan edi.



1-rasm. Tarqatilgan xotira arxitekturalari

1. Arxitektura tendentsiyalari o'zgarishi bilan, umumiy xotira SMP lari tarmoqlar orqali birlashtirilib, gibrid taqsimlangan xotira/umumiy xotira tizimlari yaratildi.

2. MPI implementatorlari o'z kutubxonalarini ikkala turdagi asosiy xotira arxitekturalarini uzluksiz boshqarish uchun moslashtirdilar. Shuningdek, ular turli xil o'zaro bog'liqliklar va protokollarni boshqarish usullarini moslashtirdilar/ishlab chiqdilar.



2-rasm. Asosiy xotira arxitekturalari

1. Bugungi kunda MPI deyarli har qanday apparat platformasida ishlaydi:

- ✓ Tarqatilgan xotira
- ✓ Umumiy xotira
- ✓ Gibrid

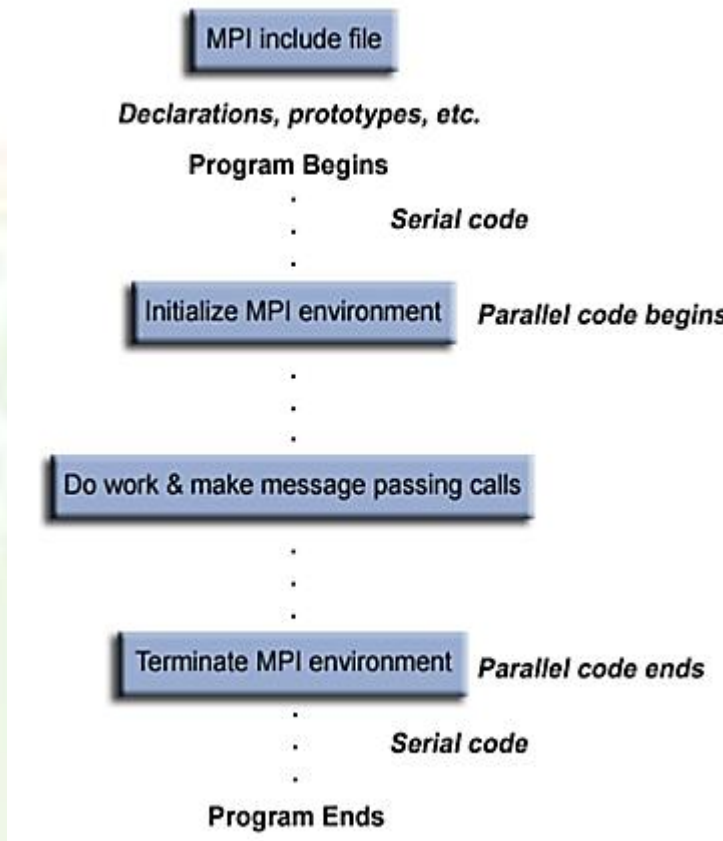
MPI dan foydalanish sabablari: Standartlashtirish - MPI standart deb hisoblanishi mumkin bo'lgan yagona xabar uzatish kutubxonasidir. U deyarli barcha HPC platformalarida qo'llab-quvvatlanadi. Amalda, u avvalgi barcha xabar uzatish kutubxonalarini almashtirdi.

Portativlik - Ilovangizni MPI standartini qo'llab-quvvatlaydigan (va unga mos keladigan) boshqa platformaga portlaganingizda manba kodingizni o'zgartirishga deyarli hojat yo'q yoki umuman ehtiyoj yo'q.

Ishlash imkoniyatlari - Sotuvchining implementatsiyalari samaradorlikni optimallashtirish uchun mahalliy apparat xususiyatlaridan foydalanish imkoniyatiga

ega bo'lishi kerak. Har qanday implementatsiya optimallashtirilgan algoritmlarni ishlab chiqish uchun bepul.

Funksionallik - MPI-3 da 430 dan ortiq protseduralar belgilangan, ular orasida MPI-2 va MPI-1 dagilarning aksariyati bor. Mavjudligi - Ham sotuvchi, ham ommaviy domen bo'lgan turli xil amalga oshirishlar mavjud. MPI dasturining umumiy tuzilishi.



3-rasm. MPI dasturining umumiy tuzilishi.

Tishli dasturlash modeli va texnologiyalari

Mavzularni yaratishning uchta sababi:

Shuning uchun, nima uchun biz bir nechta oqimlarni xohlamasligimiz aniq. Xuddi shunday, birinchi navbatda oqimlarni abadiy yaratishning cheklangan miqdordagi sabablari bor: Bir nechta protsessorlardan foydalanish. Bu klassik sabab. Agar sizning dasturiy ta'minotigiz odatda ko'p protsessorli yoki ko'p yadroli mashinada ishlayotgan bo'lsa va sizda haqiqatan ham bitta protsessor o'z vaqtida bajara oladiganidan ko'proq ish bo'lsa, bu ishlov berishni bir nechta mustaqil bajarish oqimlariga bo'lish mantiqan to'g'ri keladi.

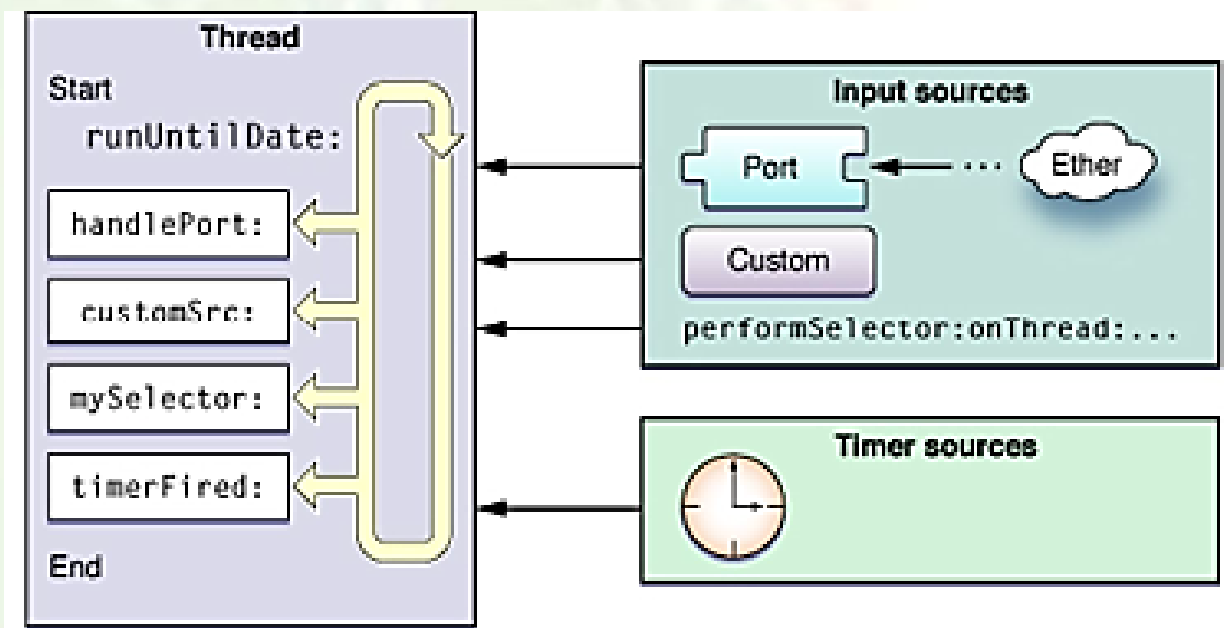


Muayyan ishlov berishni UI oqimidan olib tashlash. Bu ancha pragmatik sabab. Oddiy Windows dunyosida siz odatda oqimlarni yaratasisiz, shunda ma'lum bir fon jarayoni UI ni bo'g'ib qo'ymaydi. Men yaqinda yashayotgan Silverlight dunyosida bu unchalik muammo emas: Silverlight har qanday IO turini asinxron naqshga majbur qiladi va bu sizning UI oqimingizni blokirovka qilishiga yo'l qo'ymaydi. Biroq, ba'zida buning aksi muammo bo'lishi mumkin: UI oqimi shunchalik band bo'lishi mumkinki, siz fon oqimlarini aylantirishingiz kerak bo'ladi, shuning uchun muhim vazifalar (masalan, audio kodlash yoki dekodlash) etarlicha tez bajarilishi mumkin. Shunga qaramay, g'oya bir xil: siz foydalanuvchilaringizning xafa bo'lishining oldini olishni xohlaysiz, chunki ularning ilovasining muhim qismlari asta-sekin to'xtab qolgan ko'rinadi.

Murakkab asinxron chaqiruv naqshlarini soddalashtirish uchun. Agar siz aslida bir nechta protsessorlardan foydalanishingiz shart bo'lmasa, odatda oqimlarning ta'sirini butunlay boshqa mexanizm orqali olish mumkin, ya'ni boshqaruvni dasturning bir qismidan boshqasiga o'tkazish uchun hodisalardan foydalanish. (Bu Lauer/Needham ikkilikligi deb ataladi. 1979-yildagi mashhur maqolada Lauer va Needham "xabarga yo'naltirilgan" va "protseduraga yo'naltirilgan" tizimlar - o'qish hodisasiga asoslangan va ko'p oqimli - bir-birining duallari ekanligini va shuning uchun mantiqiy jihatdan ekvivalent arxitekturalar ekanligini ko'rsatdilar.) Ammo oqim sinxronizatsiyasini to'g'ri bajarish qiyin bo'lsa-da, butunlay asinxron chaqiruvlardan foydalanib dastur yozish yanada og'riqli bo'lishi mumkin. Buning uchun APIlar ko'pincha murakkab va noaniq bo'lib, ular sizning dasturingiz mantig'ini turli sun'iy chegaralar bo'ylab ajratishingizni talab qiladi. Ilovangizning murakkabligiga qarab, ular sizdan har qanday funktsiyani boshqa funktsiyalarga o'tkazishni so'rash mumkin bo'lgan va faqat keyinroq to'xtagan joyidan davom etadigan kooperativ ko'p vazifali modelni amalga oshirishingizni ham talab qilishi mumkin. Bu kabi muammolar nosozliklarni tuzatish va texnik xizmat ko'rsatishni ancha qiyinlashtirishi mumkin. Mavzular ancha murakkab, ammo siz mavzuni boshlashingiz, mavzu tugashini kutishingiz yoki biron bir resursni qulflashingiz mumkin bo'lgan joylardan tashqari, ko'p oqimli kod sinxron

bitta oqimli kodga juda o'xshash ko'rinadi. Va bu deyarli har doim yaxshi narsa. Yugurish sikllari mavzular bilan bog'liq asosiy infratuzilmaning bir qismidir.

Ishga tushirish sikli - bu ishni rejalashtirish va kiruvchi hodisalarni qabul qilishni muvofiqlashtirish uchun foydalanadigan hodisalarni qayta ishlash sikli. Ishga tushirish siklining maqsadi - bajarilishi kerak bo'lgan ish bo'lganda threadni band qilish va hech narsa bo'lmaganda threadni uyqu holatiga qo'yishdir. Ishga tushirish siklini boshqarish butunlay avtomatik emas. Siz hali ham thread kodini ishga tushirish siklini tegishli vaqtda boshlash va kiruvchi hodisalarga javob berish uchun loyihalashingiz kerak. Cocoa va Core Foundation ikkalasi ham threadning ishga tushirish siklini sozlash va boshqarishda yordam berish uchun ishga tushirish sikli obyektlarini taqdim etadi. Ilovangiz bu obyektlarni aniq yaratishi shart emas; har bir thread, shu jumladan ilovaning asosiy thread, tegishli ishga tushirish sikli obyektiga ega. Biroq, faqat ikkilamchi threadlar o'zlarining ishga tushirish siklini aniq ishga tushirishlari kerak. Ilova freymvorklari dasturni ishga tushirish jarayonining bir qismi sifatida asosiy threadda ishga tushirish siklini avtomatik ravishda o'rnatadi va ishga tushiradi.



3-rasm. Yugurish tsiklining tuzilishi va uning manbalari

Xulosa. Tishli uzatish barcha xotirani oqimlar o'rtasida taqsimlaydi. Bu juda xavfli, chunki boshqa oqim foydalanishi mumkin bo'lgan ma'lumotlarni tasodifan o'zgartirish juda oson, bu esa jiddiy xatolarga olib keladi. Ma'lumotlarni xavfli



kirishdan ehtiyotkorlik bilan himoya qilish vazifasi dasturchi zimmasida. Bu shuningdek (odatda) barcha jarayonlarning bir xil mashinada, bir xil jismoniy xotiraga kirish imkoniyati bilan ishlashini talab qiladi. Xabar uzatish interfeysi bilan mustaqil jarayonlardan foydalanish sizga qaysi ma'lumotlar ulashilishi va qaysi biri har bir jarayon uchun shaxsiy ekanligi ustidan ko'proq nazorat qilish imkonini beradi; bir jarayonning kutilmaganda boshqa jarayonning holatini o'zgartirish xavfi deyarli yo'q. Bundan tashqari, siz aytganingizdek, xabar uzatish interfeysi alohida mashinalardagi jarayonlar o'rtasida tarmoq orqali xabarlarini uzatish uchun umumlashtirilishi mumkin .

ADABIYOTLAR.

1. Normamatov Sardor Faxriddinovich, Safarov Ulug'bek Qarshiboevich Tsifrovye individual plany raboty professorsko-podavatelskogo sostava v meditsinskom obrazovanii. monitoring i otsenka v sisteme vysshego obrazovaniya Yangi asr innovatsiyalari jurnali 1, 51-58 2026.
2. Normamatov Sardor Faxriddinovich, Rahimov Bobur Turgunovich Texnologiya va tibbiyot. diagnosticheskaya tochnost, prognozirovaniye i kachestvo meditsinskikh uslug Yangi asr innovatsiyalari jurnali 1, 43-50 2026.
3. Normamatov Sardor Faxriddinovich, Otaxanov Polvonnazir Ergashovich Iskusstvennyy intellect v meditsine i ego znachenie Journal of new asr innovations 1, 35-42 2026.
4. Normamatov Sardor Faxriddinovich, Otaxanov Polvonnazir Ergashovich Montoring avtomatizirovannyx individualnyx planov raboty professorsko-podavatelskogo sostava v sisteme meditsinskogo vyshego obrazovaniya. Yangi asr innovatsiyalari jurnali 1, 29-34 2026 .
5. TSM Normamatov Sardor Faxriddinovich, Rahimov Bobur Turg'unovich Tibbiyotda sun'iy intellekt va uning ahamiyati Yangi asr innovatsiyalari jurnali 1, 8-15 2026.
6. UBS Normamatov Sardar Faxriddinovich , Rahimov Babur Turgunovich Medical oliy ta'lim tizimda professor o'qituvchilar avtomatlashtirilgan shaxsiy ish rejalar Yangi asr innovatsiyalari monitoringi jurnali 1, 3-7 2026.



7. N.S. Faxriddinovich, SU. Qarshiboyevich, XJ. Muzaffar o'g'li Tibbiyotda sun'iy intellekt texnologiyalari. Diagnostika aniqligi, prognoz va xizmat ko'rsatish sifati Yangi asr innovatsiyalari jurnali 93 (1), 16-23 2026
8. RB Turg'unovich, N.S. Faxriddinovich, J.Z. Ravshanovna Ta'limda raqamli transformatsiya davrida bo'lajak mutaxassislarni tayyorlashda tibbiyot va biotibbiyot muhandisligida axborot texnologiyalarining roli Web of Agriculture: Jurnal of Agriculture and Biological Sciences 2 (6), 1-8 2024.
9. S Normamatov, U Safarov, P Otoxanov, Innovatsion ta'lim texnologiyalaridan foydalangan holda fundamental fanlarni o'qitishning Qoraboyev algoritmi 2023 .
10. SF Normamatov, A. Qoraboyev Tibbiyotda axborot texnologiyalarini innovatsion texnologiyalar yordamida o'qitish metodologiyasi Umumjahon fanlarida Yevrosiyo tadqiqotlari 2023
11. S Normamatov, Z Jo'rayeva, P Otxonov Tibbiyot oliy ta'lim muassasalarida axborot texnologiyalarini o'qitish metodikasi 2023-yil.
12. S Normamatov, Z Jo'rayeva, P Otoxanov Oliy tibbiyot ta'lim muassasalarida axborot texnologiyalarini o'qitish 2023.
13. S Normamatov, U Safarov, P Otaxonov, A Qoraboyev Klinik qarorlar qabul qilishda sun'iy intellektni qo'llash Zamonaviy Amerika muhandislik, texnologiya va innovatsiya jurnali 1 (2 ...
14. S Normamatov, S Sobirjonova, U Safarov, P Otaxanov, A Koraboyev sun'iy intellektga asoslangan klinik qarorlarni qo'llab-quvvatlash tizimlari . yangi o'zbekiston tibbiyot jurnali. 2026.
15. S Normamatov, U Safarov, M Mirzahakimov, O Rozmurodov Sun'iy intellekt yordamida yurak-qon tomir kasalliklarini bashorat qilish . Yangi O'zbekiston tibbiyot jurnali.
16. N Sardor, I Farhod, M Dilmurot Kompyuter modellashtirish orqali farmatsevtika tadqiqotlarini tezlashtirish texnologiyalari Zamonaviy Amerika muhandislik, texnologiya va innovatsiya jurnali 1.



17. R Bobur, B Muratali, S Abdusamad, J Ziyoda. Tibbiyot universitetlarida fundamental fanlarni o'qitishda raqamli texnologiyalarning ahamiyati. Amerika tibbiyot va tibbiyot fanlari jurnali. 1 2023
18. AUM Abdujabbarova, AZ Sobirjonov, KD Latipova. Tibbiyot talabalariga biofizikani o'qitishning xususiyatlari. Britaniya Global Ekologiya va Barqaror Taraqqiyot Jurnali. 1 2023
19. UM Abdujabborova, AZ Sobirjonov, FS To'xtaxo'jaeva. Turli dinlarda diniy ong va axloqiy normalarni asoslash. Ta'lim fanlari bo'yicha akademik tadqiqotlar, 59-63 1 2022
20. A.Z.Sobirjonov. Abu Rayhon Beruniyning "Saydana" asarining farmakologiyadagi roli. Ta'lim fanlari bo'yicha akademik tadqiqotlar, 335-339.