

8-MA'RUZA. OPERATSION TIZIMLARDA JARAYONLAR VA OQIMLAR (THREADS)

Operatsion tizimlarda jarayonlar va oqimlar dasturlarni boshqarish va bajarishda hal qiluvchi rol o'ynaydigan asosiy tushunchalardir. Keling, jarayonlar va oqimlar o'rtasidagi asosiy xususiyatlar va farqlarni ko'rib chiqaylik.

Jarayonlar:

Jarayon - bu xotira, fayllar va tizim holatini o'z ichiga olgan resurslar to'plamidan tashkil topgan bajaruvchi dastur.

Har bir jarayon alohida va mustaqildir, ya'ni u o'z xotirasini boshqa jarayonlar bilan baham ko'rmaydi.

Jarayonlar o'zlarining jarayonlarni boshqarish blokiga ega bo'lib, unda jarayon haqida uning holati, ustuvorligi va resurslardan foydalanish kabi ma'lumotlar saqlanadi.

Jarayonni yaratish alohida tizim chaqiruvlarini talab qiladi va har bir jarayon boshqalardan mustaqil ravishda ishlaydi.

Jarayonlar bir-biri bilan jarayonlararo aloqa mexanizmlari yordamida aloqa qiladi, ular tizim qo'ng'iroqlarini o'z ichiga oladi va kerakli tizim qo'ng'iroqlari sonini ko'paytirishi mumkin.

Jarayonni almashtirish operatsion tizim bilan o'zaro aloqani o'z ichiga oladi, bu nisbatan ko'p vaqt talab qilishi mumkin.

Oqimlar:

Oqimlar (Threads) - bu jarayon ichidagi bajarish birligi. Bitta jarayonda bir nechta oqimlar mavjud bo'lishi mumkin.

Oqimlar fayl, kod va ma'lumotlar segmentlarini o'z ichiga olgan jarayonda bir xil xotira maydoni va resurslarini baham ko'radi.

Oqimlar jarayonlarga nisbatan yengilroq va kamroq tizim resurslarini talab qiladi.

Oqimni almashtirish operatsion tizim bilan o'zaro aloqani talab qilmaydi, bu uni jarayonni almashtirishdan tezroq qiladi.

Jarayon ichidagi oqimlar bir xil xotira maydoniga ega bo'lgani uchun bir-biri bilan samaraliroq aloqa qilishlari mumkin.

Oqimlar uchta holatga ega: ishlayotgan, tayyor va bloklangan. Ular jarayon doirasida bir vaqtning o'zida va mustaqil ravishda bajarilishi mumkin.

Oqimni tugatish, jarayonni tugatish bilan solishtirganda tezroq.

Jarayon tushunchasi

Jarayon - bu foydalanuvchi tomonidan ishga tushirilgan dastur, jarayon buyruqlarni bajaruvchi to'plam majmui (xotirani bajarish uchun ajratish va fayllar va kiritish/chiqarish qurilmalaridan foydalaniladigan manzil va boshqalar) va operatsion tizim boshqaruvida joylashgan joriy vaqtda uni bajarilishini (registr, dastur hisoblagich, stek holati va qo'llash belgilari) tavsiflaydi. Ish paytida operatsion tizim ko'plab dasturlarni bajaradi: paketli vazifalar, vaqtni taqsimlash rejimida foydalanuvchi dasturlari, tizim dasturlari va jarayonlari. Foydalanuvchi dasturlarini tavsiflovchi bir nechta o'xshash atamalar (terminlar) mavjud: jarayon (process), vazifa (job), topshiriq (task). Ushbu atamalarni sinonim deb hisoblashimiz mumkin.

Jarayonning muhim xususiyati: bu ketma-ket bajarilishi kerak bo'lgan hisoblash birligi, ya'ni, har bir jarayon o'zining ketma-ket boshqarish oqimiga (control flow) ega - bu jarayon tomonidan bajariladigan buyruqlar ketma-ketligidir.

Operatsion tizim tomonidan yaratilgan va boshqariladigan jarayon quyidagi asosiy ma'lumotlarni o'z ichiga oladi:

⌘ dastur hisoblagichi (program counter - PC) - bajarilayotgan joriy buyruq manzili; odatda qurilmalarning maxsus tizim registrida saqlanadi;

⌘ Stek - bu operatsion tizim tomonidan jarayon protseduralarining lokal ma'lumotlari, ularning parametrlari (argumentlari) va hisob-kitoblarni tashkil qilish uchun zarur bo'lgan ma'lumotlar o'rtasida saqlanadigan, jarayonni yaratishda operatsion tizim tomonidan ajratilgan asosiy xotiraning doimiy maydoni. Keyingi protsedurani ishga tushirishda, faollashtirish yozuvi (activation record) stekda beriladi, shuningdek, protseduraning hozirgi avlodining lokal ma'lumotlarini saqlash uchun stek kadri (stack frame) va lokal ma'lumotlar maydoni (local data area) deb ataladi. ⌘ Ma'lumotlar bo'limi (data section) – bu operatsion tizim tomonidan global o'zgaruvchilar, massivlar, tuzilmalar, obyektlar saqlanadigan jarayonga ajratilgan asosiy xotiraning statik (doimiy ravishda ajratilgan, doimiy o'lchamdagi) maydoni. Jarayonning bajariladigan kodi (buyruqlar) dastlab ikkilamchi xotirada (diskda) saqlanadi va unga kirishda to'liq yoki qisman asosiy xotiraga yuklanadi.

Jarayon holatlari Jarayon ishga tushganda, jarayon o'z holatini quyidagicha o'zgartirishi mumkin:

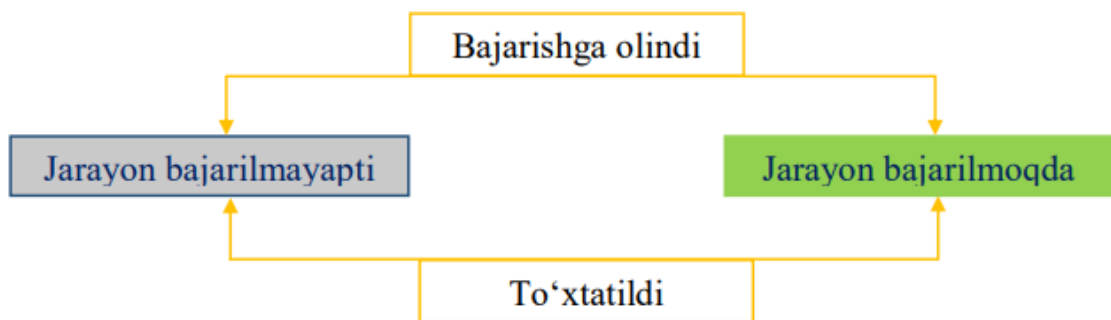
⌘ Yangi (new): jarayon operatsion tizim tomonidan yaratilgan, ammo hali ishga tushmagan;

⌘ Bajarish (running): Jarayon buyruqlari protsessorida bajariladi yoki OT ishlaydigan kompyuter tizimining protsessorida;

⌘ Kutish (waiting): jarayon qandaydir hodisa sodir bo'lishini kutmoqda, masalan, kiritish/chiqarishni (I/O) tugallanishini. Kutish holatida jarayon protsessorni egallamaydi;

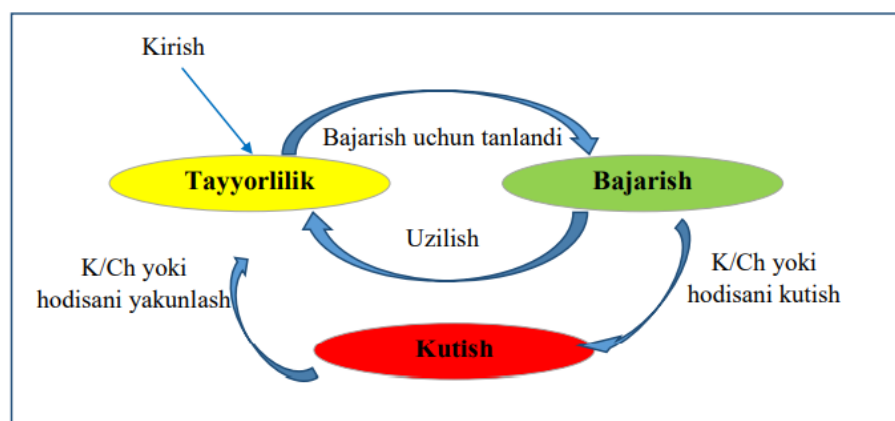
⌘ Tayyorlilik (ready): jarayon bajarilishi uchun protsessor resurslarini olishni kutmoqda. Jarayon odatda u yaratilganda ham, yoki kiritish/chiqarish (I/O) tugagandan so'ng ham (kutish holatidan) bajarilishga tayyor holatga keladi;

⌘ Tugatish (finished): Jarayonni bajarish tugallandi. Jarayon holat diagrammasi Kiritilgan topshiriqlarni hisoblash tizimlarda bajarilishi (faqat foydalanuvchi dasturlari emas, balki operatsion tizimning belgilangan qismi bo'lishi mumkin) jarayon majmuini tashkil qiladi. Shubxasiz, har bir vaqt momentida bir protsessorli kompyuter tizimida faqat bitta jarayon bajarilishi mumkin. Multidasturli hisoblash tizimlarida protsessoridagi bir jarayonni boshqasiga ulash orqali bir nechta jarayonlarni parallel qayta ishlanadi. Bitta jarayon bajarilgunga qadar qolganlari o'z navbatini kutadi. Har bir jarayon kamida ikki holatdan iborat bo'lishi mumkin: bajariluvchi jarayonlar va bajarilmaydigan jarayonlar. 2.1- rasmda jarayon holat diagrammasi modeli ifodalangan.



2.1- rasm. Jarayon holat diagrammasi modeli

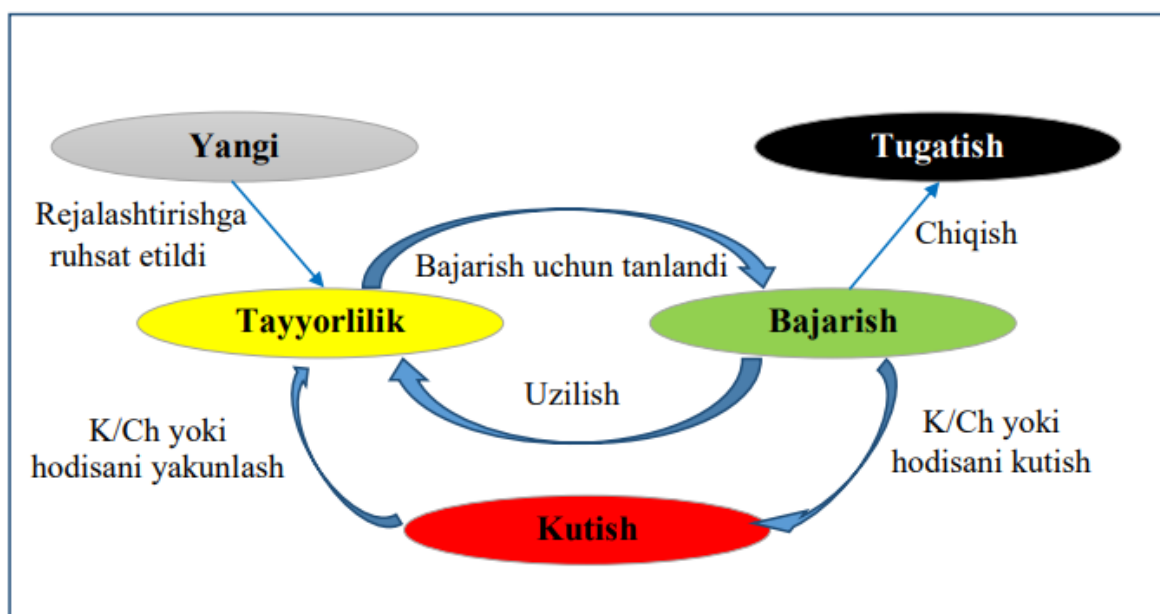
Bajariluvchi jarayon holatida joylashgan jarayon ma'lum vaqtdan so'ng operatsion tizimni tugatishi yoki ma'lum vaqtga to'xtashi va bajarilmaydigan jarayon holatiga o'tishi mumkin. Ma'lum vaqtga to'xtash jarayoni ikki sabab tufayli yuz berishi mumkin: ishni davom ettirish uchun qandaydir harakat (masalan, kiritish/chiqarish operatsiyasini yakunlash) yoki ushbu jarayonda ishlash uchun operatsion tizimda ajratilgan vaqt oralig'ining tugashi. Shundan so'ng belgilangan algoritm bo'yicha operatsion tizim bajarilmaydigan jarayon holatida joylashgan jarayonlardan birini bajarish uchun Bajarishga olindi Jarayon bajarilmayapti Jarayon bajarilmoqda To'xtatildi 39 bittasini tanlaydi. Va uni bajariluvchi jarayon holatiga o'tkazadi. Tizimda paydo bo'lgan yangi jarayon dastlab bajarilmaydigan jarayon holatida joylashadi. Bu juda ham qo'pol model bo'lib bajarish uchun jarayonlarni tanlashda u ma'lum vaqtga to'xtashi va real holatda bajarish uchun tayyor bo'lmasligi bois kutish holatida bo'lishi mumkin. Bajarilmaydigan jarayon holatidan chiqib ketish uchun ikkita yangi holatdan foydalaniladi: tayyorlilik va kutish (2.2- rasm).



2.2- rasm. Tayyorlilik va kutish holatlari

Tizimda paydo bo'lgan har qanday jarayon tayyorlilik holatida bo'ladi. Operatsion tizim tayyor jarayonlardan birini tanlab foydalanadigan algoritmni rejalashtiradi va uni bajarish holatiga o'tkazadi. Bajarish holatida jarayonni dasturiy kodini bevosita bajarishga o'tkaziladi. Bu jarayon holatidan chiqish uchun uchta sabab bo'lishi mumkin: ∞ operatsion tizim uning mavjudligini tugatishi; ∞ ayrim voqealar sodir bo'lmasligi uchun u o'z ishini davom ettirmasligi mumkin va

operatsion tizim uni kutish holatiga o'tkazadi; ∞ uni tayyorlilik holatiga qaytishda hisoblash tizimlarida uzilishlarning paydo bo'lishi natijasida (masalan, bajarilish uchun ajratilgan vaqtni tugab qolishi sababli vaqtni uzilishi). Jarayon kutish holatidan tayyorlilik holatiga o'tgandan so'ng u qaytadan bajarish uchun tanlanishi mumkin. 2.3- rasmda yana ikkita jarayon holati kiritilgan: yangi (jarayon tug'ilishi) va tugatish (bajarishni tugallanishi). Tayyorlilik Bajarish Kutish Kirish Uzilish Bajarish uchun tanlandi K/Ch yoki hodisani yakunlash K/Ch yoki hodisani kutish



2.3- rasm. Tug'ilish va bajarishni yakunlash

Endilikda jarayon hisoblash tizimida paydo bo'lishi uchun tug'ilish holati orqali o'tishi kerak bo'ladi. 2.3- rasmdan ko'rinib turibdiki, tizimda yaratilgan yangi jarayon rejalash-tirishga ruhsat etish bosqichidan o'tadi. OT tizimdagi barcha jarayonlarning navbatiga kiradi, shundan so'ng OT uni bajarishga tayyor holatga keltiradi. Shuni belgilaymizki, bajarishga tayyor bo'lgan jarayonlar navbati – jarayonlarni boshqarish uchun eng ko'p ishlatiladigan tizim tuzilmalaridan biri. Bajarilishga tayyor holatdan bajarilish holatiga o'tish jarayoni rejalashtirish asosida protsessorga vaqt kvantini ajratish natijasida OT rejalashtiruvchisi tomonidan uzatiladi. Jarayon bajarilish paytida to'xtatilishi mumkin (taymer tomonidan, xatolik natijasida va hokazo) va uzilishni qayta ishlaganidan so'ng, operatsion tizim yana ishlashga tayyor holatga qaytadi. Agar jarayonda sinxron kiritish-chiqirish amalga oshirilsa yoki jarayon biron bir hodisa sodir bo'lishini kutishi kerak bo'lsa (masalan, ma'lum bir vaqtda), jarayon kutish holatiga o'tadi. Kiritish/chiqarish (I/O) tugallanganda yoki kutilayotgan hodisa yuz berganda, protsessor vaqt kvantini darhol qabul qilmaydi, lekin bajarishga tayyor holatga o'tadi. Jarayon dasturi tugashi bilan jarayon tugallangan holatga kiradi, masalan, exit(c) tizim chaqirig'ining natijasida, c - chiqish kodi. Agar c=0 bo'lsa, jarayon muvaffaqiyatli yakunlandi deb hisoblanadi. Masalan,

Windows NT operatsion tizimi Yangi Tayyorlilik Tugatish Bajarish Kutish Rejalashtirishga ruhsat etildi Chiqish Uzilish Bajarish uchun tanlandi K/Ch yoki hodisani kutish K/Ch yoki hodisani yakunlash 41 uchun jarayon holati modeli 7 ta, Unix operatsion tizimi esa 9 ta turli xil jarayon holati modelini o'z ichiga oladi. Jarayon ichidagi operatsiyalar va ular bilan bog'liq tushunchalar Jarayon bir holatdan boshqa bir holatga mustaqil ravishda o'ta olmaydi. Jarayon holatlarini o'zgartirish, operatsion tizimga kiradigan operatsiyalarda amalga oshiriladi. Bizning modelimiz bunday operatsiyalarning soni hozirchalik holat diagrammasidagi strelkalar soniga teng. Operatsiyani uchta juft bilan bog'lash mumkin:

- ⊗ jarayonni yaratish – jarayonni tugatish;

- ⊗ jarayonnima'lumbirvaqtgato'xtatish (bajarilish holatidan tayyorlilik holatiga o'tish) – jarayonni ishga tushirish (tayyorlilik holatidan bajarilish holatiga o'tish).

- ⊗ Jarayonni bloklash (bajarilish holatidan kutish holatiga o'tish) – bloklangan jarayonni ochish (kutish holatidan tayyorlilik holatiga o'tish). Algoritmni rejalashtirish haqida gapirar ekanmiz, bizning modelimizda o'z juftiga ega bo'lmagan yana bir operatsiya paydo bo'ladi: jarayon muximligini o'zgartirish. Operatsiyalarni tashkil qilish va tugatish jarayoni bir martalik bo'ladi. Va u jarayondan faqat bir marotaba foydalaniladi (hisoblash tizimi jarayonida ayrim tizimlar hech qachon ishlashdan to'xtamaydi). Jarayon holatini o'zgartirish bilan bog'liq qolgan barcha operatsiyalar ko'p martalik hisoblanadigan ishga tushirish yoki bloklash bo'ladi.

Jarayonni boshqarish tizim osti

Asosiy tushunchalar: Jarayon tizim resurslaridan foydalanishda ba'zi talabnomalarni aniqlashi mumkin.

- ⊗ Multidasturli operatsion tizimlar bir vaqtda bir nechta jarayonlarni bajarishi mumkin.

- ⊗ Jarayonlar ko'pincha foydalanuvchi tashabbusi va uning ilovalari bo'yicha paydo bo'lishi mumkin, va bu foydalanuvchi jarayoni deyiladi.

- ⊗ O'zining funksiyalarini bajarish uchun operatsion tizim tashabbusi bilan bajariladigan jarayonlar tizim jarayoni deb ataladi.

- ⊗ Jarayonga ajratilgan operatsion tizim tezkor xotirasi sohasi majmui, manzil maydoni deb ataladi. Jarayonni boshqarish tizim ostining asosiy funksiyasi:

- ⊗ Jarayonni yaratish va yo'q qilish (jarayon bilan bog'liq bo'lgan ma'lumotlar tuzilishi);

- ⊗ Resurslarga jarayon talabi navbatini qo'llab-quvvatlash;

- ⊗ Birgalikda resurslardan foydalanishga tashkil qilingan boshqa jarayonlardan joriy jarayonga ajratilgan resurslarni himoya qilish;

- ⊗ Ayrim jarayonlarni to'xtalishi va yangidan paydo bo'lishini ta'minlash;

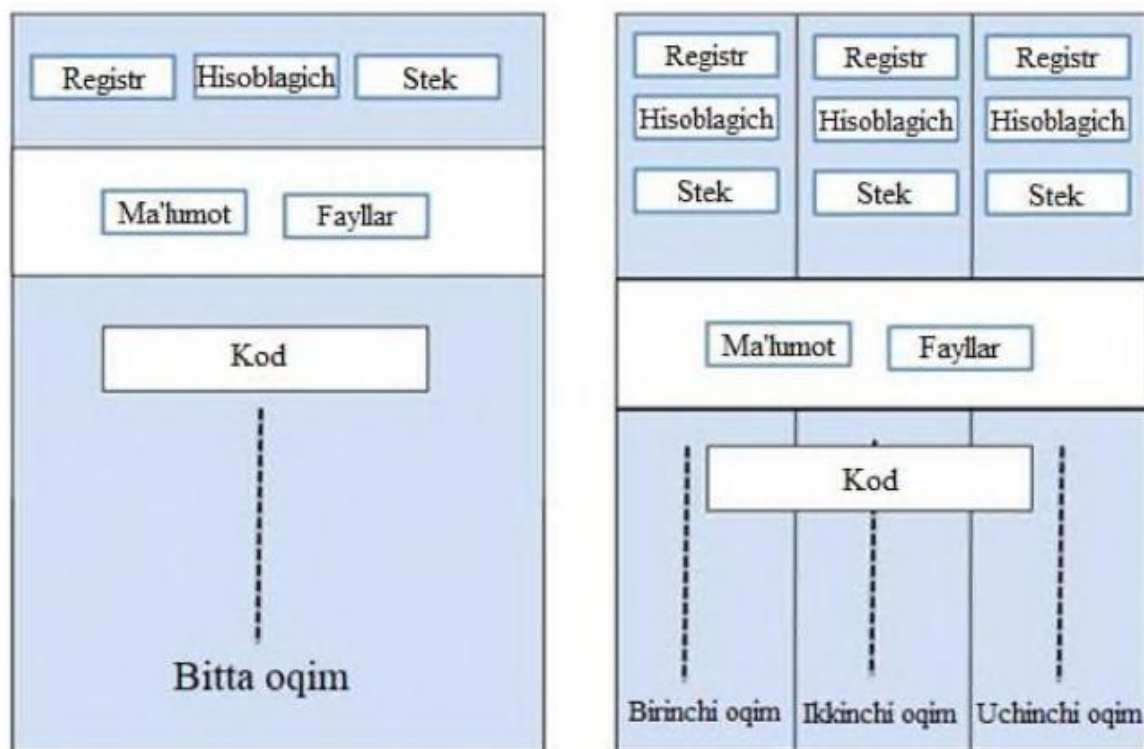
⌘ Jarayonlarni bog'lanish vositasini taqdim etish. Jarayonni boshqarish bloki Jarayonni boshqarish bloki (Process Control Block – PCB) – bu har bir jarayon bilan bog'liq quyidagi ma'lumotlarni o'z ichiga olgan jarayonni boshqarish uchun foydalanadigan tizim ma'lumotlari tuzilishi hisoblanadi:

- ⌘ Jarayon identifikator raqami (ID si);
- ⌘ Jarayon holati;
- ⌘ Buyruq hisoblagichining joriy qiymati (jarayonni bajarilishini davom ettirishda foydalaniladi);
- ⌘ Protessor registri qiymatlari (jarayonni qayta tiklashda ham foydalaniladi);
- ⌘ Protessorni rejalashtirish uchun ma'lumot (jarayon stekiga ko'rsatgich, jarayon raqami);
- ⌘ Xotirani boshqarish uchun ma'lumot (jarayon xotira maydonining chegaralari);
- ⌘ Resurslar (jarayon bajarilishi uchun ajratilgan);
- ⌘ Hisob ma'lumotlari (e'lon qilingan ijro vaqtidan qolgan jarayonning umumiy bajarilish vaqti, jami kiritish/chiqarish vaqti va boshqalar);
- ⌘ Kiritish/chiqarish holatlari to'g'risidagi ma'lumotlar (ochiq fayllar ro'yhati).

Operatsion tizimlarda oqim (threads)lar

Operatsion tizimlarda oqim (threads)lar tushunchasi. Dasturlarning ko'p oqimli (multi threading) bajarilishi,

Operatsion tizimdagi oqimlar va ularning turlari Oqim – bu operatsion tizimning mohiyati, protsessorda ko'rsatmalar to'plamini bajarish jarayoni, aniqrog'i dastur kodi. Oqimlarning umumiy maqsadi ikki yoki ko'plab turli xil vazifalarni protsessorda parallel bajarishdir.



2.5- rasm. Bitta oqimli bitta jarayon, uchta oqimli bitta jarayon ko‘rinishi

Oqim o‘zining kod segmenti, ma’lumotlar segmenti va ochiq fayllarini teng huquqli oqimlari bilan almashadi. Agar bir oqim kod segmentining xotira elementini o‘zgartirsa, qolgan barcha oqimlar buni ko‘radi. Oqim yengil jarayon deb ham ataladi. Oqimlar parallellash orqali dastur samaradorligini oshirish usulini taqdim etadi. Quyidagi rasmda bir oqimli va ko‘p oqimli jarayonlarning ishlashi ko‘rsatilgan. Oqimlar - bu klassik jarayonga mos keladigan ishchi oqimlarni kamaytirish hisobiga operatsion tizimning ish faoliyatini yaxshilashga qaratilgan dasturiy yondashuv.

Jarayon va oqim o‘rtasidagi farq Jarayon Oqim

1 Jarayon og‘ir yoki ko‘p resurs talab qiladi Oqim yengil va jarayonga qaraganda kamroq resurslarni talab qiladi

2 Jarayonni almashtirish operatsion tizim bilan o‘zaro aloqani talab qiladi Oqimlarni almashtirish operatsion tizim bilan o‘zaro aloqani talab qilmaydi

3 Bir nechta qayta ishlash muhitida har bir jarayon bir xil kodni bajaradi, ammo o‘z xotirasi va fayl resurslariga ega Barcha oqimlar farzand jarayonini bir xil ochiq fayllar to‘plamini bo‘lishishi mumkin

4 Agar bitta jarayon bloklangan bo‘lsa, unda birinchi jarayon blokdan chiqarilmaguncha boshqa jarayon bajarilmaydi Bitta oqim bloklangan va kutayotgan paytda, xuddi shu vazifadagi ikkinchi oqim ishlashi mumkin

5 Bir nechta jarayonlar oqimlardan foydalanmasdan ko'proq resurslardan foydalanadi
Ko'p oqimli jarayonlar kamroq resurslardan foydalanadi

6 Bir nechta jarayonlarda har bir jarayon mustaqil ravishda amalga oshiriladi Bitta oqim boshqa oqim ma'lumotlarini o'qishi, yozishi yoki o'zgartirishi mumkin Har bir oqim aniq bitta jarayonga tegishli va jarayondan tashqarida hech qanday oqim mavjud bo'lmaydi.

Har bir oqim alohida boshqaruv oqimini anglatadi. Oqimlar tarmoq serverlari va veb-serverlarni amalga oshirishda muvaffaqiyatli ishlatilgan. Ular shuningdek, umumiy xotiraga ega bo'lgan ko'p protsessorlarda ilovalarni parallel ravishda bajarish uchun munosib asosni ta'minlaydi.

- ⊗ Oqim afzalliklari ⊗ Oqimlar kontekstni almashtirish vaqtini kamaytiradi;

- ⊗ Oqimlardan foydalanish jarayon ichida parallellikni ta'minlaydi;

- ⊗ Samarali aloqa;

- ⊗ Kontekst oqimlarini almashtirish va yaratishda yanada tejamkor;

⊗ Oqimlar ko'p protsessorli arxitekturalardan keng miqyosda va samarali foydalanishga imkon beradi. Oqim turlari Oqimlar quyidagi ikki usulda amalga oshiriladi:

- ⊗ Foydalanuvchi darajasidagi oqimlar (User Level Threads) - foydalanuvchi boshqaradigan oqimlar;

- ⊗ Yadro darajasidagi oqimlar (Kernel Level Threads) – yadroda ishlaydigan operatsion tizim boshqaradigan oqimlar, operatsion tizim yadrosi. Foydalanuvchi darajasidagi oqimlar Bunday holda, oqimlarni boshqarish yadrosi oqimlarning mavjudligi haqida xabardor bo'lmaydi. Oqimlar kutubxonasi oqimlarni yaratish va yo'q qilish, xabarlar va ma'lumotlarni oqimlar o'rtasida uzatish, oqimlarni bajarishni rejalashtirish va oqim kontekstlarini tiklash va saqlash kodlarini o'z ichiga oladi.

Foydalanuvchi darajasidagi oqimlarning afzalliklari:

- ⊗ Oqimni almashtirish yadro rejimi imtiyozlarini talab qilmaydi;

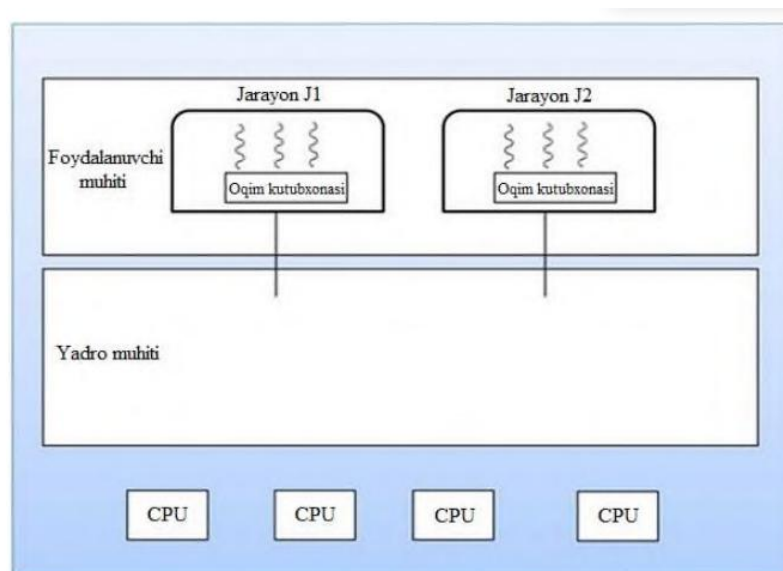
- ⊗ Foydalanuvchi darajasidagi oqim har qanday operatsion tizimda ishlashi mumkin;

- ⊗ Rejalashtirish foydalanuvchi darajasidagi oqimga xos bo'lgan dastur bo'lishi mumkin;

- ⊗ Foydalanuvchi darajasidagi oqimlar tezda yaratiladi va boshqariladi. Foydalanuvchi darajasidagi oqimlarning kamchiliklari:

- ⊗ Oddiy operatsion tizimda tizim qo'ng'iroqlarining aksariyati bloklanadi;

- ⊗ Ko'p oqimli ilovalar ko'p protsessorli ishlov berishning afzalliklaridan foydalana olmaydi.



2.6- rasm. Foydalanuvchi darajasidagi oqimlar

Yadro darajasidagi oqimlar Bunday holda, oqimni boshqarish yadro tomonidan amalga oshiriladi. Ilova sohasida oqimlarni boshqarish kodi mavjud emas. Yadro oqimlari to‘g‘ridan-to‘g‘ri operatsion tizim tomonidan qo‘llabquvvatlanadi. Har qanday ilovani ko‘p oqimli dasturlash mumkin. Ilovadagi barcha oqimlar bitta jarayonda qo‘llab-quvvatlanadi. Yadro butun jarayon uchun va jarayon tarkibidagi alohida oqimlar uchun kontekst ma’lumotlarini saqlab turadi. Yadro tomonidan rejalashtirish oqimlar asosida amalga oshiriladi. Yadro oqimlarni yaratish, rejalashtirish va boshqarishni yadro sohasida amalga oshiradi. Yadro oqimlari odatda foydalanuvchi oqimlariga qaraganda sekinroq yaratiladi va boshqariladi.

Afzalliklari ⚭ Yadro bir vaqtning o‘zida bir xil jarayondan bir nechta jarayonga bir nechta oqimlarni rejalashtirishi mumkin;

⚭ Agar jarayonda bitta oqim bloklangan bo‘lsa, yadro xuddi shu jarayonda boshqa oqimni rejalashtirishi mumkin;

⚭ Yadro modulini o‘zi ko‘p oqimli qilishi mumkin; Kamchiliklari

⚭ Yadro oqimlari odatda foydalanuvchi oqimlariga qaraganda sekinroq yaratiladi va boshqariladi;

⚭ Xuddi shu jarayon davomida boshqaruvni bitta oqimdan boshqasiga o‘tkazish, yadro rejimiga o‘tishni talab qiladi.

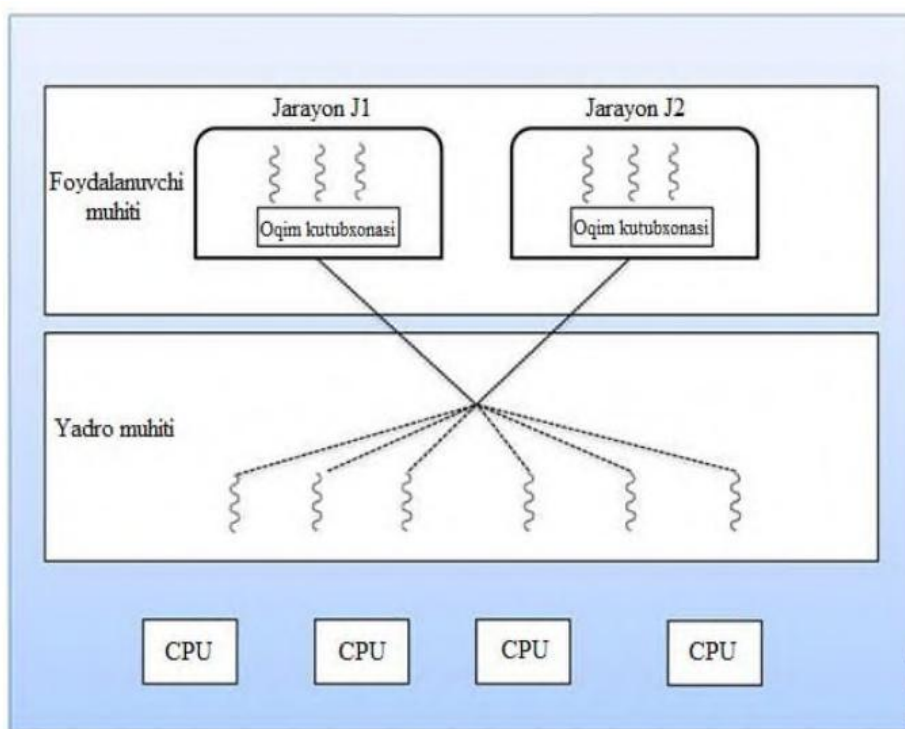
Ko‘p oqimli model Ba’zi bir operatsion tizimlar birlashtirilgan foydalanuvchi darajasidagi oqimlar va yadro darajasidagi oqimlarni birlashtiradigan vositani ta’minlaydi. Solaris ushbu birlashtirilgan yondashuvga yaxshi misol hisoblanadi. Birlashtirilgan tizimda bitta dasturdagi bir nechta oqimlar bir nechta protsessorlarda parallel ravishda ishlashi mumkin, va bloklash tizim chaqiruvi butun jarayonni bloklamasligi kerak.

Ko‘p oqimli modellar uch turga bo‘linadi:

⌘ Many to many modeli;

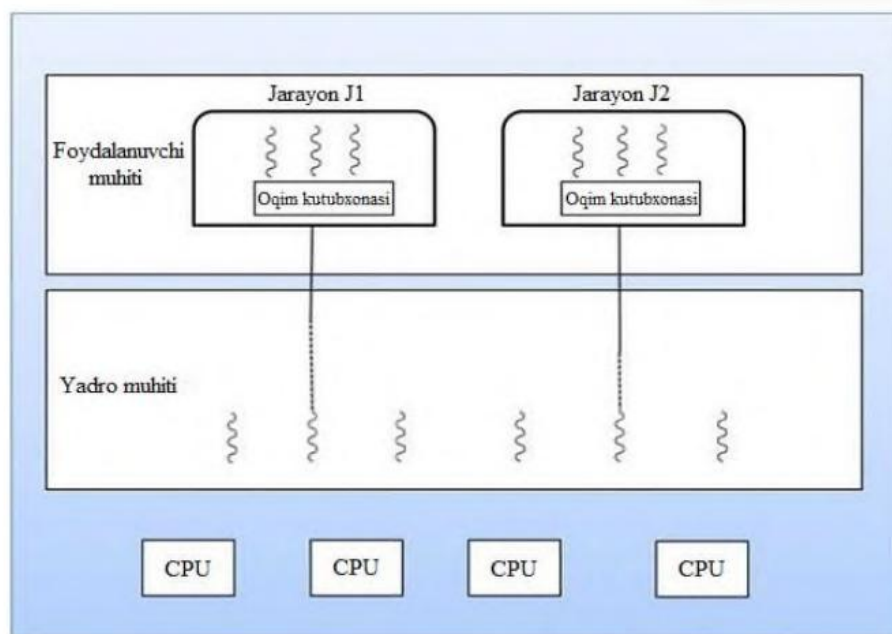
⌘ Many to one modeli;

⌘ One to one modeli. Many to many modeli Many to many modeliga misol quyidagi 2.7- rasmda keltirilgan. Many to many modeli har qanday foydalanuvchi oqimlarini teng yoki kichik sonli yadro oqimlariga ko'paytiradi. 2.7- rasmda ko'p oqimli model ko'rsatilgan, unda 6 foydalanuvchi darajasidagi oqimlar 6 yadro darajasidagi oqimlar bilan ko'paytiriladi. Ushbu modelda ishlab chiquvchilar, qancha foydalanuvchi oqimlari kerak bo'lsa, shuncha yaratishi mumkin va mos keladigan yadro oqimlari ko'p protsessorli mashinada parallel ravishda ishlashi 48 mumkin. Ushbu model parallellash bilan eng yaxshi aniqlikni ta'minlaydi, va oqim bloklash tizim chaqirig'i amalga oshirganda, yadro yana bitta oqimni bajarishi mumkin.



2.7- rasm. Many to many modeli

Many to One Modeli Many to one modeli ko'p foydalanuvchi darajasidagi oqimlarni bitta yadro darajasidagi oqimlar bilan taqqoslaydi. Oqimlarni boshqarish foydalanuvchi maydonida oqimlar kutubxonasi tomonidan amalga oshiriladi. Agar oqim bloklash tizim chaqiruvini amalga oshirsa, butun jarayon bloklanadi. Bir vaqtning o'zida faqat bitta oqim yadroga kirishi mumkin, shuning uchun ko'p protsessorli tizimlarda bir nechta oqimlar parallel ravishda ishlay olmaydi. Agar foydalanuvchi darajasidagi oqimlar kutubxonalari operatsion tizimda tizim ularni qo'llab-quvvatlamaydigan tarzda amalga oshirilsa, unda yadro oqimi many to one modelini qo'llaydi.



2.8- rasm. Many to One modeli

One to One Modeli Foydalanuvchi darajasidagi oqim va yadro darajasidagi oqim o'rtasida one to one munosabatlar mavjud. Ushbu model many to one modelga qaraganda ko'proq parallellashni ta'minlaydi. Bundan tashqari, u bloklash tizim chaqirig'i amalga oshirilganda, boshqa oqimni ishga tushirishga imkon beradi. Mikroprotssessorlarda parallel bajarilishi uchun bir nechta oqimlarni qo'llab-quvvatlaydi. Ushbu modelning kamchiligi shundaki, foydalanuvchi oqimini yaratish uchun yadroga mos keladigan oqim kerak bo'ladi. OS/2, Windows NT va Windows 2000 OTlari one to one modelidan foydalanadi.

Nazorat savollari

1. Oqim nima?
2. Oqimning jarayondan farqini tushuntiring.
3. Oqimni afzalliklarini keltirib o'ting.
4. Oqim turlari va bir-biridan farqlari?
5. Foydalanuvchi darajasidagi oqimlarni tushuntiring.
6. Yadro darajasidagi oqimlarni tushuntiring.
7. Ko'p oqimli model turlarini keltiring.
8. Many to many modelini tushuntiring.
9. Many to one modelini tushuntiring.
10. One to one modelini tushuntiring.
11. Operatsion tizimda jarayon nima?
12. Jarayonlarni boshqarish nima uchun kerak?
13. Jarayon dastur hisoblagichi nima uchun kerak?
14. Jarayonlarni boshqarish bloki nima va u nima uchun kerak?
15. Jarayonlarni boshqarish blokining vazifasi nima?
16. Jarayonlarni boshqarish bloki o'z ichiga qanday ma'lumotlarni oladi?
17. Jarayon qanday holatlarda bo'lishi mumkin?
18. Yangi holat bo'yicha ma'lumot bering.