9-MA’RUZA. OPERATSION TIZIMLARDA JARAYONLARNING BOSHQARILISHI

Multidasturli muhitda OT protsessor qaysi jarayonni, qachon va qancha vaqt olishini hal qiladi. Bu funksiya jarayonni rejalashtirish deb nomlanadi. Operatsion tizim protsessorni boshqarish uchun quyidagi amallarni bajaradi:

ϖ Protsessor va jarayon holatini kuzatadi;

ϖ Jarayonga protsessorni (CPU) ajratadi;

ϖ Jarayon talab qilinmasa, protsessorni o‘chiradi. Protsessorni (CPU ni) rejalashtirish multidasturli operatsion tizimlarning asosidir. Protsessorni jarayonlar o‘rtasida almashtirish orqali operatsion tizim kompyuterni unumdorligini oshirishi mumkin. 64 Protsessorni rejalashtirish va dispetcherlashtirish operatsion tizimning eng muhim funksiyalaridan biri hisoblanadi. Ushbu bo‘limda quyidagi masalalar ko‘rib chiqilgan:

ϖ Jarayonlarni rejalashtirishning asosiy tushunchalari;

ϖ Dispetcherlashtirish mezonlari;

ϖ Dispetcherlashtirish algoritmlari (FCFS, SJF, RR va boshqalar); ϖ Bir necha jarayonlarni dispetcherlashtirish;

ϖ Real vaqtda dispetcherlashtirish;

ϖ Ko‘p darajali navbatlar.

Jarayonlarni rejalashtirish Kompyuter ko‘p vazifalik rejimda ishlayotganda markaziy protsessorda bir vaqtning o‘zida bir nechta jarayon yoki oqimlar ishga tushadi. Agar bunday vaziyatda tayyorlilik holatda bo‘lsa bir vaqtning o‘zida ikki yoki undan ko‘p jarayon yoki oqimlarga duch kelinadi. Agar faqat bitta protsessorga kirish ruhsat etilgan bo‘lsa, bu jarayonlardan qaysi biri birinchi bajarilishi tanlaniladi. Ko‘plab operatsion tizimlarda bu tanlov rejalashtirish deb ataladi, u foydalanadigan algoritm esa rejalashtirish algoritmi deb ataladi. Jarayonlarni rejalashtirishni qo‘llash, shuningdek bir biridan farqli bo‘lgan ayrim oqimlarni rejalashtirishga qo‘llash mumkinmi degan ko‘plab bir xil bo‘lgan savollar mavjud. Yadro oqimni boshqarsa, u holda odatda har bir oqimni, u aynan qaysi jarayonda yotganiga qaramay rejalashtiriladi. Dastlab biz jarayonlar va oqimlar qanday

rejalashtirishni ko‘rib chiqamiz. Shundan so‘ng biz oqimlarni rejalashtirish va qator paydo bo‘ladigan noyob savollarni ko‘rib chiqamiz. Magnitli lentada keltirilgan perfokart ko‘rinishli shakldagi ma’lumotlarni kiritishni amalga oshirgan paketli tizim davrida rejalashtirish algoritmlari soddaroq bo‘lgan: faqat lentadagi navbatdagi topshiriqni ishga tushirish talab qilingan. Ko‘p topshiriqli tizimlarning paydo bo‘lishi rejalashtirish algoritmlarini takomillashtirdi, ushbu holatda odatda xizmat ko‘rsatishni kutayotgan bir nechta foydalanuvchiga, bir vaqtning o‘zida xizmat ko‘rsatiladi. Ayrim universal mashinalar haligacha vaqtni ajratish rejimida topshiriqlarni paketli topshiriq usulida hisoblaydi, va rejalashtiruvchi navbatdagi bajariladigan ish qanday bo‘lishini hal qilishi kerak bo‘ladi: paketli topshiriqni bajaradi yoki terminalda o‘tiruvchi foydalanuvchi bilan interaktiv aloqani ta’minlaydi. Bunday vaqtli protsessorli mashinalari uchun resurslar yetishmovchiligi bo‘ladi, kuchli rejalashtiruvchi mashinalar unumdorligini his qilishi va foydalanuvchilarni qanoatlantira olishi mumkin. Shaxsiy kompyuterlarning paydo bo‘lishi vaziyatni ikki holatga yo‘naltirdi. Birinchisi, asosiy vaqtni bitta faol jarayonga qaratdi. Foydalanuvchi hujjatga matnni kiritishda protsessor bir vaqtning o‘zida fonli rejimda dasturni kompilatsiyalagan. Foydalanuvchi protsessor matni uchun buyruqni terganda, rejalashtiruvchi qaysi jarayoni ishga tushirishni tanlamagan, chunki matnli protsessor yagona nomzod bo‘lgan. Kompyuterlarning ikkinchi davrida, kompyuterlar tez ishlashi natijasida, resurslar yetishmovchiligi kuzatilgan. Shaxsiy kompyuterlar uchun ko‘plab dasturlar foydalanuvchiga kiruvchi axborotlar belgilangan (chegaralangan) tezlikda taqdim etiladi, lekin bu tezlik bilan emas. Markaziy protsessor uni qayta ishlashiga bog‘liq. Endilikda xatto bir vaqtning o‘zida bir nechta dasturlar bilan ishlash imkoniga ega bo‘lindi, masalan, matnli jarayon va elektron jadval. Shu sababli shaxsiy kompyuterlarda jarayonlarni rejalashtirish muxim rol o‘ynamaydi.

Albatta mavjud ilovalar, markaziy protsessorning barcha resurslaridan foydalanadi: masalan, bir soatli yuqori sifatli videoda har bir 108 000 kadrni (NTSC da) yoki 90 000 kadrni (PAL da) rangli gammalarda to‘g‘irlash uchun katta quvvatga ega sanoat hisoblash kompyuterlari talab qilinadi, lekin ilova qoidalari bundan

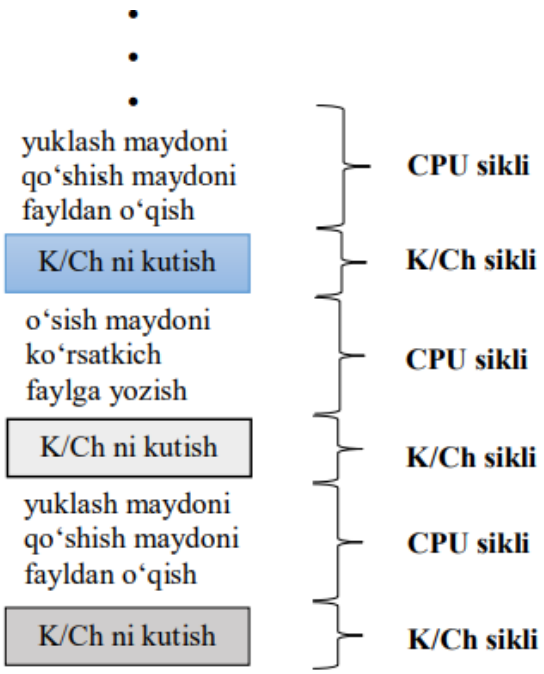
mustasno. Tarmoq xizmatlari talab qilinganda vaziyat o‘zgaradi. Bu yerda protsessor vaqtida bir nechta jarayonlarga qarshi kurashiladi, shu sababli yangi topshiriqlar olinadi. Masalan, markaziy protsessorda ishga tushirilgan jarayonlardan birontasini tanlash zarurati paydo bo‘lganda kunlik statistikani to‘playdi va foydalanuvchi so‘rovlariga xizmat ko‘rsatish jarayonlaridan birini tanlaydi (agar navbatdagi jarayonga muximlilik berilsa, foydalanuvchi o‘zini yaxshi xis qiladi). “To‘g‘ri” jarayonni tanlashdan tashqari rejalashtiruvchi markaziy protsessorning to‘g‘ri ishlashi haqida o‘ylashi kerak. Dastlab foydalanuvchi rejimida yadro rejimiga o‘tkazilishi kerak. So‘ng ularni keyingi qayta yuklanish uchun uning registeri jarayon jadvalida saqlashni o‘z ichiga olgan joriy jarayon holati saqlanishi kerak.

Shundan so‘ng, keyingi jarayonni tanlash uchun rejalashtirish algoritmi ishga tushiriladi. So‘ng yangi jarayon xotira kartasiga muvofiq xotirani boshqarish blokiga yuklanadi. Va nihoyat yangi jarayon ishga tushiriladi. Qachon rejalashtiriladi Rejalashtirishning kalit so‘zi qarorni qabul qilish vaqti bo‘ladi. Rejalashtirish talab qilinadigan turli xil holatlar mavjud. Birinchidan, jarayonlarning qaysi biri birinchi bajarilishini aniqlashtiruvchi yangi jarayonni yaratish.

Har ikkala jarayon tayyorlilik holatida joylashganligi sababli, rejalashtiruvchi jarayonning onasi (parent process) yoki uning farzandlarini (child process) tanlashga asoslanib qaror qabul qilishi kerak bo‘ladi. Ikkinchidan, rejalashtiruvchi jarayon tugayotganda qaror qabul qilishi kerak bo‘ladi. Jarayon boshqa bajarilmaydi (endilikda u mavjud bo‘lmaydi), shu bois bajarishga tayyor bo‘lgan jarayonlar ichidan birortasini tanlash kerak bo‘ladi. Agar jarayonlar bajarishga tayyor bo‘lmasa, odatda tizim bo‘sh jarayonni ishga tushuradi. Uchinchidan, jarayon belgiga yoki biron bir boshqa jarayonni tanlash asosida kiritish/chiqarish operatsiyasini yakunlanishini kutishda bloklansa jarayonni bajarish uchun boshqa biron bir jarayonni tanlash kerak bo‘ladi. Ba’zida bu rolni bloklash bajaradi. Masalan, A muxim rolni o‘ynayapti va B jarayonni paydo bo‘lishini kutayotgan bo‘lsa, u holda A jarayon ishlashni davom ettirishga imkon berish uchun uni tang sohasidan ushbu jarayonni chiqishiga imkon berib B jarayonga bajarish navbatini taqdim etadi. Lekin murakkab tarafi, odatda rejalashtiruvchi ma’lumotlarga bog‘liq

bo‘lgan zarur axborotlarga ega bo‘lmaydi. To‘rtinchidan, rejalashtiruvchi kiritish/chiqarishda to‘xtalishlar paydo bo‘lganda qaror qabul qilishi kerak bo‘ladi. Agar kiritish/chiqarish operatsiya yakunlangan bo‘lsa, kutishida bloklangan qandaydir jarayonda o‘z ishini yakunlab kiritish/chiqarish qurilmasida to‘xtalish bo‘lib o‘tgandan so‘ng, ishlashga tayyor bo‘lishi mumkin.

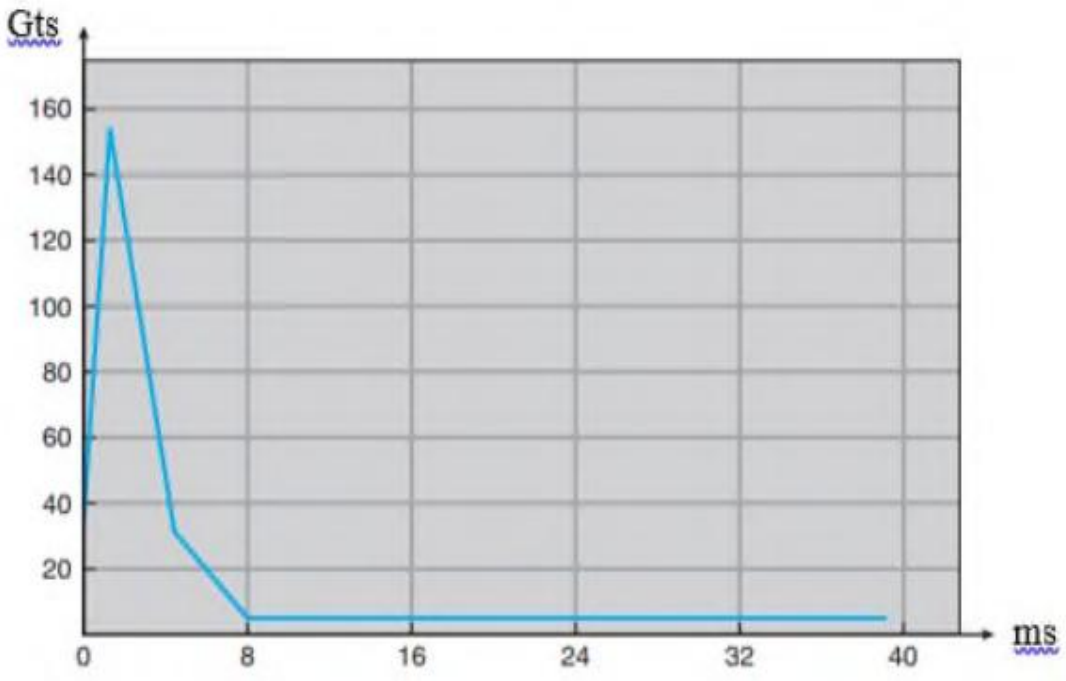
Rejalashtiruvchi qandaydir jarayonni ishga tushirish bo‘yicha qaror qabul qilishi kerak bo‘ladi. Protsessorni rejalashtirish Protsessorni rejalashtirish tizimdagi jarayonlar orasida uning vaqtini taqsimlash hisoblanadi. Rejalashtirishdan maqsad - multidasturlash yordamida erishiladigan protsessorning maksimal yuklanishini ta’minlashdan iborat. Istalgan jarayonning bajarilishiga CPU / K/Ch sikli – protsessordan foydalanish va kiritish/chiqarishni kutish davrlarining navbatlashishi sifatida qarash mumkin.



2.16- rasm. CPU va K/Ch sikllari o‘zgaruvchan ketma-ketligi

Bir protsessorli tizimda bir vaqtning o‘zida faqat bitta jarayon bajarilishi mumkin. Boshqa jarayonlar esa protsessorni bo‘shashini kutishi kerak. Multidasturlashning vazifasi – protsessordan maksimal darajada foydalanishdir. Ushbu turdagi rejalashtirish opeatsion tizimning asosiy funksiyasidir. Deyarli barcha kompyuter resurslari foydalanishdan oldin rejalashtiriladi. Protsessor, shubhasiz kompyuterning asosiy resurslaridan biridir. Shunday qilib, uni rejalashtirish

operatsion tizimni loyihalashda muhim o‘rin tutadi. Protsessorni rejalashtirishning muvaffaqiyati jarayonlarning kuzatiladigan xususiyatlariga bog‘liq: jarayonning bajarilishi protsessorni bajarish siklidan va K/Ch ni kutishdan iborat. Jarayonlar ushbu ikki holat orasida o‘zgarib turadi. 2.17- rasmda operatsion tizimlardagi jarayonlarning haqiqiy holatini tahlil qilish asosida, protsessor faoliyati davrlarining taxminiy gistogrammasi ko‘rsatilgan. yuklash maydoni qo‘shish maydoni fayldan o‘qish o‘sish maydoni ko‘rsatkich faylga yozish • • • yuklash maydoni qo‘shish maydoni fayldan o‘qish K/Ch ni kutish K/Ch ni kutish K/Ch ni kutish CPU sikli CPU sikli CPU sikli K/Ch sikli K/Ch sikli K/Ch sikli



2.17- rasm. Protsessor faoliyati davrlarining taxminiy gistogrammasi Diagramma shuni ko‘rsatadiki, faoliyat davri qanchalik qisqa bo‘lsa, bunday davrlarning chastotasi shunchalik yuqori va aksincha, faoliyat davrlarining chastotasi ularning davomiyligiga teskari proporsionaldir. Protsessorni rejalashtiruvchi Rejalashtiruvchi - xotiraga yuklangan va bajarishga tayyor bir nechta jarayonlardan birini tanlaydigan va ulardan biri uchun protsessorni ajratadigan OT komponenti hisoblanadi. Rejalashtirish bo‘yicha yechimlar quyidagi hollarda qabul qilinishi mumkin, agar jarayon:

1. Bajarilish holatidan kutish holatiga qayta ulanadi (masalan, jarayon kiritish/chiqarish qurilmasi ishini tugashini kutsa).

2. Bajarilish holatidan bajarishga tayyorlik holatiga o‘tishga qayta ulanadi (masalan, uzilish sodir bo‘lganda).

3. Kutish holatidan tayyorlik holatiga qayta ulanadi (masalan, kiritish/chiqarish hodisasi tugashi bilan).

4. Yakunlanadi. 1- va 4- turlardagi rejalashtirish jarayonni uzmasdan rejalashtirish (non-preemptive) atamasi bilan belgilanadi. 2- va 3- turlardagi rejalashtirish jarayonni uzish bilan rejalashtirish (preemptive) atamasi bilan belgilanadi. Protsessor menejeri 2 quyi menejerdan tashkil topadi:

1. Vazifani rejalashtiruvchi;

2. Jarayonni rejalashtiruvchi.

1. Vazifani rejalashtiruvchi Vazifani rejalashtiruvchi yuqori darajadagi rejalashtiruvchi hisoblanadi va uning funksiyalari quyidagicha:

ϖ Vazifalarni kirish navbatidan tanlab oladi;

ϖ Ularni xarakteristikasiga qarab jarayonlar navbatiga kiritadi;

ϖ Maqsad: vazifalarni tizim resurslaridan maksimal foydalanish tartibida joylashtirish;

ϖ Resurslarni doimiy band holatda saqlashni tashkillashtiradi.

ϖ Kiritish/chiqarish va hisoblash o‘rtasidagi o‘zaro muvozanatni ta’minlaydi.

2.Jarayonni rejalashtiruvchi Jarayonni rejalashtiruvchi past darajadagi rejalashtiruvchi hisoblanadi va uning funksiyalari quyidagicha:

ϖ Jarayonlarni bajarilishi uchun protsessorni taqsimlaydi;

ϖ Protsessor resurslarini qachon va qancha muddatga olishini belgilaydi;

ϖ Uzilishlarni qayta ishlashni hal qiladi;

ϖ Qachon jarayon to‘xtatilishini va qayta tiklanib prtosessordan foydalanishini tashkil qiladi.

Vazifa va jarayon holatlari Vazifani rejalashtiruvchi yangi (new) va tugatish (finished) holatlariga, jarayonlarni rejalashtiruvchi esa tayyorlilik (ready), bajarilish

(running) va kutish (waiting) holatlarini boshqaradi. Jarayon operatsion tizimdan vazifa rejalashtiruvchisiga uzatilganida uning (jarayonning) holati har doim yangi (new) sifatida o‘rnatiladi. Barcha jarayonlar dastlab vazifani rejalashtiruvchiga kelib tushadi. Qachonki jarayon vazifa rejalahtiruvchisidan jarayon rejalashtiruvchisiga o‘tkazilganda uning holati tayyorlilik (ready) holatiga o‘zgartiriladi. Agar protsessor ishlash uchun tayyor bo‘lsa (bo‘sh bo‘lsa) barcha kelayotgan jarayonlarni ko‘zdan kechiradi va ma’lum algoritmlar asosida ularni tanlaydi va xotirani bo‘sh deb hisoblab, jarayonni bajarishni boshlaydi. Jarayonning holati bajarilayotgan (running) ga 70 o‘zgartiriladi. Oldindan belgilangan vaqtdan so‘ng jarayon to‘xtatiladi va boshqa bir jarayon (bajarilish uchun) protsessorni egallaydi. To‘xtatilgan jarayonning holati jarayon rejalashtiruvchisi tomonidan tayyorlilik (ready) holatiga o‘zgartiriladi. Jarayonlarning bunday almashtirilishi oldindan rejalashtirish siyosati deb nomlanadi. Agar jarayon ishga tushirilganda foydalanuvchi yoki boshqa jarayon tomonidan kiritish/chiqarish amali bajarilishini kutsa, u holda jarayon kutish (waiting) holatiga o‘tkaziladi. Kiritish/chiqarish jarayoni yakunlanganligi haqida xabar kiritish/chiqarish qurilmasidan jarayonni rejalashtiruvchiga yetkazilganda jarayonning holati tayyorlilik (ready) holatiga o‘zgartiriladi.

Nihoyat, jarayon yakunlanganida yoki xatolik yuz berganida jarayon (oldindan) tugatiladi va uning holati tugatishga (finished) o‘tkaziladi. Odatda, jarayon holatining o‘zgarishlari jarayonni rejalashtiruvchi tomonidan amalga oshiriladi, va vazifani rejalashtiruvchiga bu o‘zgarishlar haqida axborot beriladi. Shundan so‘ng vazifani rejalashtiruvchi holatni tugatishga (finished) o‘zgartiradi. Protsessor dispetcheri Protsessor dispetcheri protsessorni rejalashtiruvchi tanlagan jarayonga beradigan OT komponenti hisoblanadi.

Dispetcher quyidagi amallar ketma-ketligini bajaradi:

ϖ Kontekstni (tarkibni) qayta ulaydi;

ϖ Protsessorni foydalanuvchi rejimiga qayta ulaydi;

ϖ Foydalanuvchi dasturini qayta yuklash uchun uni mos manzil bo‘yicha o‘tishini bajaradi. Dispetcherning yashirin aktivligi (dispatch latency) - bitta

jarayonni to‘xtatish va boshqa jarayonni boshlash uchun dispetcherga talab qilinadigan vaqt hisoblanadi. Ma’lumki, tizim bu vaqtni minimallashtirishga intilishi kerak, lekin rejalashtirish mezonlari to‘plami murakkabroq. Rejalashtirish mezonlari Tizim u yoki bu darajada hisobga olishi kerak bo‘lgan protsessorning beshta asosiy rejalashtirish mezonlari mavjud. Protsessordan foydalanish (CPU utilization) - maksimal bo‘lishi mumkin bo‘lgan vaqt davrida uni bandlik rejimida saqlash hisoblanadi.

Optimallashtirish mezoni: bu ko‘rsatkichni maksimallashtirish. Tizimning o‘tkazish qobiliyati (throughput) - vaqt birligi ichida o‘zining bajarilishini tugatadigan jarayonlar soni (o‘rtacha) hisoblanadi. Jarayonga ishlov berish vaqti (turnaround time) - qandaydir jarayonni bajarilishi uchun zarur bo‘ladigan vaqt hisoblanadi. Optimallash mezoni: bu ko‘rsatkichni minimallashtirish. Kutish vaqti (waiting time) - jarayon bajarilishga tayyor jarayonlar navbatida kutadigan vaqt hisoblanadi. Javob vaqti (response time) - interfaol tizimda vazifani bajarilish vaqti eng yaxshi mezon bo‘lmasligi mumkin. Istalgan optimallashdagi kabi strategiyaga bog‘liq bo‘lmagan holda, barcha mezonlarni bir vaqtda qoniqtirish mumkin emas. Quyida turli rejalashtirish algoritimlarini (yoki strategiyalarini) ko‘rib chiqamiz va ko‘rsatilgan mezonlarning optimalligiga erishish nuqtai nazaridan ularning afzalliklari va kamchiliklarini ko‘rib chiqamiz.

Jarayonlarni rejalashtirish algoritmlari First-Come-First-Served (FCFS) algoritmi First-Come-First-Served (kelish tartibida xizmat ko‘rsatish, ya’ni, birinchi kelganga birinchi xizmat ko‘rsatish (FIFO) kabi bir xil) – algoritmi eng oddiy rejalashtirish algoritmi bo‘lib, bunda protsessorning resurslari jarayonlarga ular iste’mol qiladigan resurslarga, xususan, jarayonning bajarilishi uchun talab qilinadigan u bildirgan vaqtga bog‘liq bo‘lmagan holda tizimga kelishi (kirishi) tartibida taqdim etiladi. Bu va boshqa algoritmlarni ko‘rib chiqishda jarayonlarning nomlari va ularning qandaydir vaqt birliklarida ifodalanadigan bajarilish vaqt diapazonlarini Gant diagrammalaridan (Gantt charts) foydalanib aniqlaymiz. Quyidagi misolni ko‘rib chiqamiz. J1, J2 va J3 jarayonlar quyidagi aktivliklar davrlari bilan ko‘rsatilgan tartibda tizimga kiritilgan bo‘lsin:

Aktivlik davri J1 24 J2 3 J3 3

U holda ularni rejalashtirish uchun FCFS algoritmidan foydalanishda protsessorni birinchi bo‘lib uzoq bo‘lishiga qaramasdan, birinchi jarayonni oladi. Bu holda protsessorni jarayonlar orasida taqsimlanishi (1- misol)

Shunday qilib, kutish vaqti J1 = 0; J2= 24; J3 = 27 bo‘ladi. O‘rtacha kutish vaqti: (0 + 24 + 27)/3 = 17 Agar jarayonlar tartibi boshqacha - J2, J3, J1 bo‘lsa (tizimga oxirgi kiritilgan jarayon – eng uzoq), u holda ularni rejalashtirish natijasi mutlaqo boshqacha bo‘ladi

FCFS algoritmi bo‘yicha rejalashtirish sxemasi (2- misol) Bu holda jarayonlarni kutish vaqti: J1 = 6; J2 = 0; J3 = 3. O‘rtacha kutish vaqti: (6 + 0 + 3)/3 = 3 Bu natija oldingi natijaga qaraganda ancha yaxshi. Birinchi misol namoyish etgan natija samarasi (convoy effect) – qisqa jarayon uzoq jarayondan keyin xizmat ko‘rsatiladigan hollarda jarayonlarni o‘rtacha kutish vaqtini ortishi deyiladi. Shortest Job First (SJF) algoritmi Shortest Job First (SJF, dastlab eng qisqa vazifani bajarish) algoritmi protsessorni rejalashtirish algoritmi bo‘lib, bunda protsessor birinchi navbatda tizimdagi mavjud jarayonlardan eng qisqasiga beriladi. Bu holda har bir jarayon bilan uning navbatdagi aktivlik davri davomiyligi bog‘lanadi. Bu davomiylik eng qisqa jarayonga birinchi xizmat ko‘rsatilishi uchun ishlatiladi. Bu algoritmni qo‘llanishining ikkita sxemalari bo‘lishi mumkin:

1. Jarayonlarni uzmasdan – jarayonga protsessor berilayotgan vaqtda uning vaqt kvanti tugamasdan jarayon uzilmasligi kerak.

2. Jarayonlarni uzish bilan – agar aktivlik vaqti aktiv jarayonning qolgan vaqtidan kichik bo‘lgan yangi jarayon kelsa, aktiv jarayonni to‘xtatish.

Bu sxema Shortest-Remaining-Time-First (SRTF – dastlab eng qisqa vaqt) nomi bilan ma’lum. Ko‘rish qiyin emaski, SJF algoritmi u berilgan jarayonlar to‘plami uchun minimal o‘rtacha kutish vaqtini ta’minlashi mazmunida optimal bo‘ladi. Jarayonlarni uzmasdan SJF algoritmining qo‘llanishiga misolni ko‘rib chiqamiz. Jarayonlar to‘plami, tizimda ularning paydo bo‘lishi vaqtlari va ularning aktivligi vaqtlari quyidagicha:

Aktivlik vaqti J1 0.0 7 J2 2.0 4 J3 4.0 1 J4 5.0 4 Jarayonlarni uzmasdan SJF algoritmi bo‘yicha jarayonlarni rejalashtirish sxemasi Bu holda o‘rtacha kutish vaqti = (0 + 6 + 3 + 7)/4 = 4. Endi o‘sha jarayonlarga uzilishli SJF algoritmini qo‘llaymiz va o‘rtacha kutish vaqti qanday o‘zgarishini tahlil qilamiz. Algoritmning qo‘llanishi natijasi

Bu holda tizimga qisqaroq jarayon tushishi momentida jarayonning uzilishi prinsipi bir necha marta qo‘llanadi: 2 momentda 1- jarayon uziladi va qisqaroq 2-jarayon bajarila boshlanadi, 4 momentda 2- jarayon uziladi va qisqaroq 3- jarayon bajarila boshlanadi. Diagrammadan ko‘rinib turibdiki, jarayonlarning uzilishi prinsipining qo‘llanishi tufayli protsessordagi jarayonning uzluksiz bajarilishi davrlari yonma-yon bo‘lishi va boshqa jarayonlarni bajarilishi davrlarini bilan o‘rin almashishi mumkin. Bu holda o‘rtacha kutish vaqti = (9 + 1 + 0 +2)/4 = 3, ya’ni kutilganidek, u jarayonlarni uzilishi prinsipi qo‘llanilmasligiga qaraganda kichik bo‘ldi. Ustuvorliklar bo‘yicha rejalashtirish Bu algoritmda har bir jarayon bilan uning ustuvorligi (butun son) bog‘lanadi. Protsessor eng katta ustuvorlikli jarayonga beriladi (kichik son yuqoriroq ustuvorlikni bildiradi, ya’ni jarayonning eng yuqori ustuvorligini 1 ga teng deb olamiz). Bu algoritm oldingi algoritm kabi uzilishli va uzilishsiz variatlarga ega. Shu bilan birga, SJF algoritmiga ustuvorliklar bo‘yicha rejalashtirish sifatida qarash mumkin, unda navbatdagi aktivlik vaqti ustuvorlik hisoblanadi. Ustuvorliklar bo‘yicha rejalashtirishda "ochlik" (starvation) muammosi – past ustuvorlikli jarayonlar hech qachon bajarilmasligi va cheksiz kutadigan vaziyatlar vujudga keladi. Operatsion tizimlarda bu muammoni yechishning an’anaviy usuli jarayonning ortishini hisobga olish (aging) hisoblanadi. Vaqt o‘tishi bilan jarayonning ustuvorligi tizim orqali oshiriladi. Round Robin (RR) algoritmi Round Robin (RR, halqali tizim) algoritmi bu barcha jarayonlarga navbat bo‘yicha bir xil vaqt kvantlarini berish hisoblanadi. Algoritmning nomi AQShdagi ommaviy qarta o‘yinidan kelib chiqadi. Bu algoritmda har bir jarayon protsessor vaqtining uncha katta bo‘lmagan kvanti – odatda 10-100 millisekundni oladi. Bu vaqt tugagandan keyin jarayon uziladi va tayyor jarayonlarni oxiriga joylashtiriladi. Agar bajarilishga tayyor jarayonlar navbatida n jarayonlar va vaqt kvanti q ga teng bo‘lsa,

u holda har bir jarayon 1/n protsessor vaqtini eng kattasi q birlikdan bir marta qismlab oladi. Hech bir jarayon (n-1) q vaqt birligidan ortiq kutmaydi. Bu algoritmning unumdorligi q koeffitsientga bog‘liq:

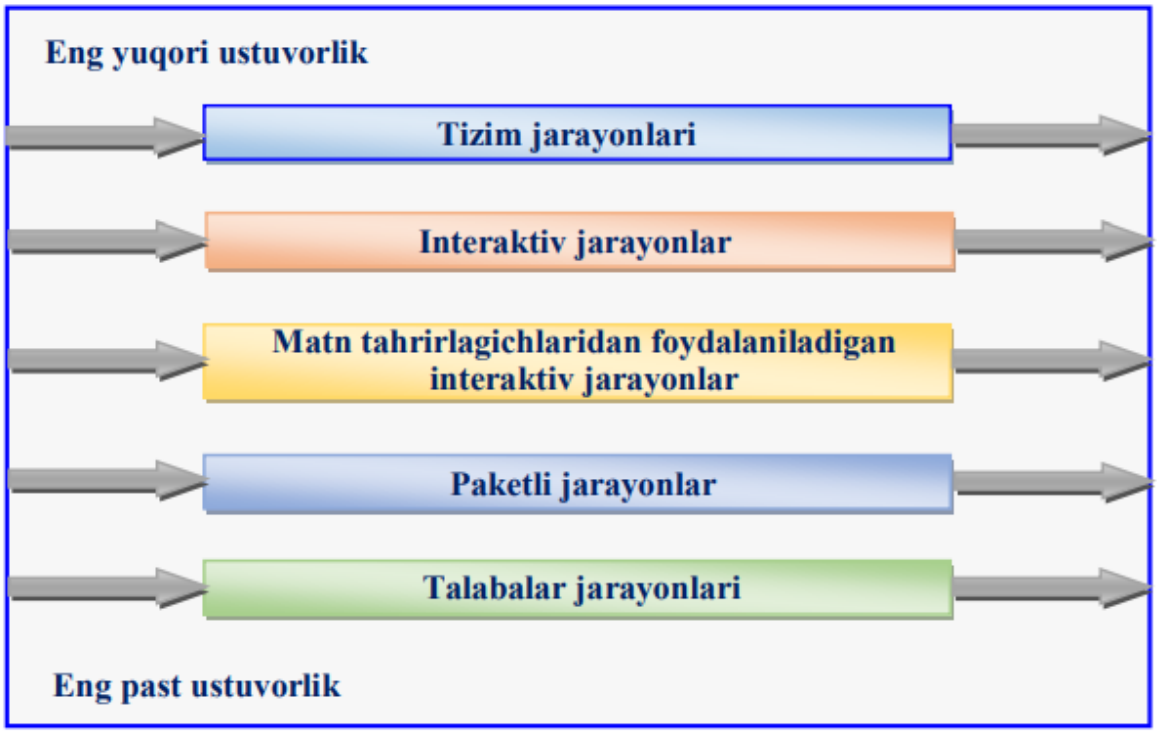
ϖ agar q yuqori bo‘lsa, u holda algoritm FCFS algoritmga deyarli ekvivalent;

ϖ agar q past bo‘lsa, u holda q tarkibiy qayta ulanish vaqtidan katta bo‘lishi kerak, aks holda bitta jarayondan boshqa jarayonga qayta ulanishga ustama sarflar o‘ta katta bo‘ladi. RR algoritmini qo‘llanilishi misolini ko‘rib chiqamiz. Tizimda aktivlik vaqtli quyidagi jarayonlar mavjud bo‘lsin J1 53 J2 17 J3 68 J4 24 q = 20 vaqt kvantili RR algoritm bo‘yicha protsessorni rejalashtirish sxemasi RR algoritmini qo‘llanishiga misol (q = 20) Odatda RR algoritmi SJF algoritmga qaraganda yomon aylanish vaqtiga ega (chunki har bir jarayon vaqt kvantlari boshqa jarayonlarga beriladigan vaqtda navbatdagi vaqt kvantini kutishi kerak bo‘ladi), lekin yaxshi javob vaqtiga ega. Protsessor kvant vaqti va kontekstni almashtirish vaqti Ko‘p darajali navbat Binobarin, tizimdagi jarayonlar turli o‘ziga xosliklarga (masalan, paketli va interaktiv) ega bo‘lishi mumkin, amalda operatsion tizimlarda bajarilishga tayyor jarayonlar navbati ikkita navbatlarga bo‘linadi:

ϖ asosiy (interaktiv jarayonlar);

ϖ fon (paketli jarayonlar). Har bir navbat o‘z rejalashtirish algoritmiga ega bo‘ladi. Asosiy navbat RR, fon navbat FCFS rejalashtirish algoritmiga ega bo‘ladi. Bu aralash algoritmda navbatlar orasidagi rejalashtirish, ya’ni u yoki bu navbatdan jarayonlarni tanlash algoritmi zarur bo‘ladi. Navbatlar orasidagi rejalashtirish quyidagi turlarga bo‘linadi:

ϖ Qayd etilgan ustuvorlikli – asosiy navbatdan, keyin fon navbatdan barcha jarayonlarga xizmat ko‘rsatish. Bunda “och qolish” ehtimolligi mavjud. ϖ Vaqt oralig‘ini ajratish – har bir navbat qandaydir protsessor vaqt oralig‘ini oladi, u jarayonlar orasida taqsimlanishi mumkin, masalan, 80% asosiy navbatdagi RRga va 20% fon navbatdagi FCFSga taqsimlanishi mumkin. 2.24- rasmda jarayonlarni rejalashtirish uchun ko‘p darajali navbat tuzilmasiga real misol keltirilgan.



2.24- rasm. Ko‘p darajali navbatni rejalashtirishga misol

Eng yuqori ustuvorlikka tizim jarayonlari ega, keyin interaktiv jarayonlar, undan past ustuvorlikka esa matn tahrirlagichlari chaqiriladigan interaktiv jarayonlarga ega (ular foydalanuvchilarning sekin ishlashi tufayli sezilarli katta vaqtni egallaydi), keyin paketli va nihoyat talabalar jarayonlari keladi. Real vaziyat shunday, lekin muallif talabalar jarayonlarini “kamsitilishini” to‘g‘ri hisoblamaydi. Aynan ularga tizim jarayonlaridan keyingi ustuvorlikni, masalan, diplom ishlarini himoya qilishdan oldingi davrda berish kerak bo‘ladi.