

ABDULLAYEVA G.X., HAMRAYEVA S.I.,
XUDAYBERGANOV T.R., XUDAYBERGANOV O.F.

OPERATION TIZIMLAR



O'ZBEKISTON RESPUBLIKASI OLIY TA'LIM, FAN
VA INNOVATSIYALAR VAZIRLIGI

MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT
AXBOROT TEXNOLOGIYALARI UNIVERSITETI
URGANCH FILIALI

ABDULLAYEVA G.X., HAMRAYEVA S.I.,
XUDAYBERGANOV T.R., XUDAYBERGANOV O.F.

OPERATION TIZIMLAR

O'zbekiston Respublikasi Oliy ta'lim, fan va innovatsiyalar
vazirligi tomonidan o'quv qo'llanma sifatida tavsiya etilgan

Toshkent
"METODIST NASHRIYOTI"
2024

UDK: 004.45(075.8)
BBK: 32.973ya7
O - 62

Abdullayeva G.X.
Operatsion tizimlar/ Hamrayeva S.I., Xudayberganov T.R.,
Xudayberganov O.F./ O'quv qo'llanma. – Toshkent:
"METHODIST NASHRIYOTI", 2024. – 176 b.

Axborot texnologiyalarining inson faoliyati barcha jabxalariga kirib borishi va tobora chuqurlanishi natijasida ularni tadbiq etish, xisoblash texnikasi bilan muloqot vositasi maxsus dasturlari operatsion tizimlari, operatsion tizimlar tarixi, operatsion tizim rivojlanish tendentsiyalari, istiqboli xamda Respublikamizdagi axborot-kommunikatsiya texnologiyalari sohasidagi islohatlar natijalari asosida yangi dasturiy tizimlar yaratish va bu tizimlarni boshqarish masalalarini qamraydi va talaba bu fanni o'zlashtirish borasida oladigan ko'nikma va malakasiga qo'yiladigan talablar keng yoritib berilgan.

Mazkur o'quv qo'llanma oily ta'lim muassasalarining 5330300 – Axborot xavfsizligi(sohalar bo'yicha), 5330500 – Kompyuter injiniringi ("Kompyuter injiniringi", "AT-Servis"), 5330600 – Dasturiy injiniringi, 5350100 – Telekommunikatsiya texnologiyalari ("Telekommunikatsiyalar", "Teleradioeshittirish"), 5350400 – Axborot – kommunikatsiya texnologiyalari sohasida kasb ta'limi ta'lim yo'nalishlari o'qituvchilari va talabalari uchun mo'ljallangan.

Taqrizchilar:

O.K. Xo'jayev - Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti Urganch filiali Axborot texnologiyalari kafedrası PhD.

M.S. Sharipov – Urganch davlat universiteti "Axborot texnologiyalari" kafedrası dotsenti.

O'quv qo'llanma O'zbekiston Respublikasi Oliy va o'rta maxsus ta'lim vazirligining 2021–yil 25-dekabrda 538-sonli buyrug'iga asosan nashr etishga ruxsat berilgan.(Ro'yxatga olish raqami 538-457).

ISBN 978-9910-03-210-3

© Abdullayeva G.X. va boshq., 2024.
© "METHODIST NASHRIYOTI", 2024.

KIRISH

O'zbekiston Respublikasi Prezidentining 2019 yil 8 oktyabrda PF-5847-son Farmoni bilan tasdiqlangan "O'zbekiston Respublikasi Oliy ta'lim tizimini 2030 yilgacha rivojlantirish kontseptsiyasi" da oliy ta'lim jarayonlariga raqamli texnologiyalar va zamonaviy o'qitish usullarni joriy etish, yoshlarni ilmiy faoliyatga keng jalb etish, korruptsiyaga qarshi kurashish, muhandislik-texnik ta'lim yo'nalishlarida tahsil olayotgan talabalar ulushini oshirish, kredit-modul tizimini joriy etish, o'quv rejalarda amaliy ko'nikmalarni oshirishga qaratilgan mutaxassislik fanlari bo'yicha amaliy mashg'ulotlar ulushini oshirish bo'yicha aniq vazifalar belgilab berilgan.

O'zbekiston Respublikasi Prezidenti Sh.M. Mirziyoevning 2020 yil 24-yanvardagi Oliy Majlisga yo'llagan murojaatnomasida 2020 yilga "Ilm, ma'rifat va raqamli iqtisodiyotni rivojlantirish yili" deb nom berishning taklif etilishi, maktab bitiruvchilarini oliy ta'lim bilan qamrab olish darajasini bosqichma-bosqich oshirib borish, ta'lim yo'nalishlari va o'qitiladigan fanlarni qayta ko'rib chiqish, mutaxassislikka aloqasi bo'lmagan fanlar sonini 2 barobar qisqartirish, oliy ta'limda o'quv jarayoni kredit-modul tizimiga o'tkazish, qator oliy ta'lim muassasalarini o'zini o'zi moliyaviy ta'minlashga o'tkazish, ta'lim sohasini to'liq raqamlashtirish, davlat-hususiy sheriklik mexanizmlarini ta'lim sohasiga ham keng tatbiq etish masalalari shu kunning dolzarb masalalari ekanligi ta'kidlab o'tilgan.

Ushbu murojaatnomada birinchi o'ringa xalqning bilim darajasini oshirish masalasi qo'yilgan, bu esa millatning raqobatbardoshligini belgilovchi omil hisoblanadi. Shu sababli, tabiiyki, ushbu hujjatda zamonaviy talablar va jahon standartlariga javob beradigan milliy ta'lim tizimini yaratish muammosining elementlari ham o'z aksini topgan.

Prezidentimiz tomonidan chiqarilgan farmonlar oliy ta'lim tizimini rivojlantirishga alohida xizmat qiladi. Shu farmonlardan kelib chiqib, hozirgi kunda oliy ta'limda o'qitiladigan fanlarga ham alohida e'tibor berilmoqda.

Yil sayin hisoblash tizimlarini qo'llash ancha kengayib bormoqda. Natijada korxonalarining katta-kichik bizni

jarayonlarining samaradorligini oshirish va butun dunyo bo'yicha oddiy foydalanuvchilarning shaxsiy kompyuter bilan ishlashini yengillashtirish imkonini bermoqda. Bu masalada Operatsion tizimlarining o'rni katta hisoblanadi. Operatsion tizim – bu oddiy foydalanuvchi kompyuteridan tortib superkompyuterlarga bo'lgan har bir kompyuterining ajralmas qismi bo'lib, tizimli dasturiy ta'minot hisoblanadi.

Bu o'quv qo'llanmada yuqoridagi masalalarni ma'lum darajada yechishga qaratilgan. Bunda zamonaviy operatsion tizimlarning ishlashi, qo'llanilishi va arxitekturasi xususiyatlarining asosiy elementlari bayon etilgan.

Qo'llanmada OT qurilishi va ishlashining umumnazariy masalalari bayon etilgan. OTning asosiy tushunchalari, OT rivojlanishining asosiy bosqichlari, OT funktsiyalari, rivojlanish davri, OTning arxitekturaviy xususiyatlari, kiritish-chiqarish funktsiyalari, jarayonlar, resurslar.

Qo'llanmaning ma'lum qismini OTda ishlashda amaliy ko'nikma olishga qaratilgan. Bunda Windows OTlar oilasida amaliy ishlar bajarishga qaratilgan.

Qo'llanma 5330500 "Kompyuter injiniringi (Kompyuter injiniringi, AT-servis)", 5330600 "Dasturiy injiniring", 5350100 "Telekommunikatsiya texnologiyalari (Telekommunikatsiyalar)", 5350400 "Axborot kommunikatsiyalari sohasida kasb ta'limi" yo'nalishlari talabalari uchun mo'ljallangan, biroq boshqa mutaxassislikdagi talabalar uchun xam, shuningdek, barcha hohlovchilar uchun xam foydali bo'lishi mumkin. OT tashkillashtirish va ishlashining asosiy printsiplarini chuqurroq o'rganish u yoki bu hisoblash tizimlarining qurilmalari va dasturiy ta'minotlarini sotib olish, undan samarali va qulay foydalanish, uni qisman yangilash va almashtirish haqida asosli ravishda qaror qabul qilish imkonini beradi.

1. OPERATSION TIZIMLAR FANINING MAZMUN VA MOHIYATI

1.1. Operatsion tizim tushunchasi

Operatsion tizim (OT) – bu foydalanuvchiga qulay tarzda kompyuter qurilmalaridan foydalanish imkonini beruvchi dasturdir[1].

OT – kompyuterining apparat vositalarini boshqarishni, fayllar bilan ishlashni, ma'lumotlarni kiritish-chiqarishni, shuningdek amaliy dasturlar va utilitalarni bajarilishini ta'minlovchi kompyuter dasturlarining asos jamlanmasidir.

Yuqorida 2 ta ta'rifdan tashqari, foydalanuvchilar OTga qator tushunchalarni beradilar:

- OT virtual mashina sifatida;
- OT resurslarni boshqarish tizimi;
- OT foydalanuvchi va dasturlar himoyachisi;
- OT doimiy ishlaydigan yadro.

Bu tushunchalarni kengroq ko'rib chiqamiz.

OT virtual mashina sifatida. Shaxsiy kompyuter arxitekturasini amaliy dasturlardan foydalanish uchun mashina buyruqlari darajasida ishlatish juda noqulaydir. Masalan, diskdan ma'lumotni o'qishda, uning ichki qurilmalarini bilishni talab etadi – diskni aylantirish buyrug'ini kiritish kontrolleri, yo'lakchalarni qidirish va formatlash, sektorlarni o'qish va yozish va sh.k.

Foydalanuvchi bilishi shart bo'lmagan bunday ortiqcha harakatlarning bajarilishini yashirish uchun nisbatan oddiy va yuqori darajali mavhumlik evaziga foydalanuvchi va kompyuter o'rtasidagi interfeysni ta'minlash talab etiladi. Masalan, diskdagi ma'lumotni o'qish yoki yozish uchun ochish, so'ngra yopish.

OT resurslarni boshqarish tizimi. Agar bitta kompyuterda ishlayotgan bir necha dasturlar bir vaqtning o'zida ma'lumotlarni chop qilish uchun printeriga uzatsa, u holda qatorlar va sahifalar qorishib ketadi. OT bunday chalkashlikni bartaraf etishi uchun chop qilishga navbat tashkil qilishi lozim. Shu kabi muhim muammolardan biri – ko'p foydalanuvchili kompyuterlar uchun resurslarni boshqarish muammosidir.

Shunday qilib, OT resurslar menejeri sifatida har xil dasturlar o'rtasida protsessornlarni, xotiralarni va boshqa resurslarni tartiblab va nazorat qilib taqsimlashni amalga oshiradi.

OT foydalanuvchi va dasturlar himoyachisi. Agar bitta kompyuterda bir nechta foydalanuvchilarni ishlashi ta'minlanishi lozim bo'lsa, u holda ularning faoliyati xavfsizligini tashkil etish muammosi yuzaga keladi. Demak, quyidagilarni ta'minlash kerak:

- diskda ma'lumotlarni saqlash, boshqa foydalanuvchilar tomonidan fayllarni zararlantirishidan va o'chirilishidan himoyalash;
- biror foydalanuvchilar dasturlarining boshqa foydalanuvchilar dasturlari ishiga aralashishiga yo'l qo'ymaslik;
- hisoblash tizimidan ruxsatsiz foydalanishlariga yo'l qo'ymaslik.

Bu masalalar foydalanuvchilar va dasturlarning xavfsiz ishlashini tashkil qilish uchun OT tarkibiga kiritilgan.

OT doimiy ishlaydigan yadro. OT bir nechta amaliy dasturlar bilan o'zaro aloqada bo'ladigan kompyuterda doimiy ishlaydigan dastur sifatida qarash kerak bo'ladi. Ma'lumki, bunday tushuncha qisman haqiqatga yaqin, ya'ni, ko'pgina zamonaviy OTlarda kompyuterda OTning bir qismi – yadrosi ishlaydi.

1.2. Kompyuter tizimlarining sinflanishi

Birinchi dasturlar bevosita mashina kodlarida yaratilgan. Buning uchun, mikroprotsessor arxitekturasini va uning asosidagi tizimni mukammal bilish zarur. Hisoblash texnikasining rivojlanishi borasida, ko'p uchraydigan amallarni (operatsiyalarni) ajratib, ular uchun dasturiy modullar yaratilib, keyinchalik ulardan dasturiy ta'minotda foydalana boshladilar. Shunday qilib, 50-chi yillarda, birinchi dasturlash tizimlarini yaratishda kiritish chiqarish amallari uchun, keyinchalik matematik amal va funktsiyalarni hisoblash uchun modullar yaratildi. Keyinchalik, rivojlanish yuqori darajadagi translyatorlarni yaratilishiga olib keldi, ya'ni operatorlar o'rniga zaruriy funktsiyalarni chaqiriqlarni qo'yish imkoniyati kelib chiqdi. Kutubxonalar soni oshib bordi. Natijada, amaliy dasturiy ta'minot ishlab chiquvchilaridan tizim arxitekturasini mukammal bilish talab qilinmay qoldi. Ular, dasturiy tizimga mos chaqiriqlar bilan murojaat

va ulardan kerakli servis va funktsiyalarni olish imkoniga ega bo'ldilar. Bu dasturiy tizim OT dir.

Zamonaviy OT asosiy tashkil etuvchilari – bu yadro, kiritish-chiqarish tizimi, komanda protsessori, fayl tizimi. Yadro masalalar va resurslarni boshqarish, sinxronlashtirish va o'zaro bog'lanishi bo'yicha asos funktsiyalarni ta'minlaydi. Komanda protsessori, komandalarni qabul qilish va ularga ishlov berish, foydalanuvchi talabi bo'yicha mos xizmatlarni chaqirishni ta'minlaydi.

Kiritish va chiqarish tizimi, Tashqi qurilmalar bilan ma'lumotlarni kiritish va chiqarish masalasini ta'minlaydi.

OT kutubxonalarida bu funktsiyalarning mavjudligi, har bir ularni dasturlash tizimi vositalari bilan har bir dasturga qo'shmaslik imkonini beradi. Dasturlash tizimlari faqat, kiritish-chiqarish tizimi kodlariga murojaatni generatsiya qiladi va kattaliklarni tayyorlaydi. KCh tizimi, kiritish-chiqarish qurilmalari turlari ko'p bo'lgani uchun eng murakkab hisoblanadi. Bunda, nafaqat samarali boshqarish, balki amaliy dasturchilarga qurilmalardan abstraktlashtirishga imkon beradigan qulay va samarali virtual interfeysni ta'minlaydi. Boshqa tomondan, parallel bajariladigan talay masalalarni kiritish-chiqarish qurilmalariga murojaatni ta'minlash talab qilinadi. Ba'zi kiritish-chiqarish dasturchilaridan ba'zilar qurilmalardan mustaqildir, ularni kiritish-chiqarish ko'pgina qurilmalariga, qo'llash mumkin.

Fayl – bir xil tuzilishga ega bo'lgan yozuvlar majmuasi ko'rinishida tashkil etilgan ma'lumotlar to'plamidir.

Fayl tizimi, foydalanuvchiga ma'lumotlar tuzilishining mantiqiy darajasi va amallar bilan ish ko'rish imkonini beradi. Fayl tizimi, ma'lumotlarni diskda yoki boshqa ma'lumot jamlanmasida tashkil qilish usulini belgilaydi.

Barcha zamonaviy OT lar o'z fayl boshqaruv tizimiga ega. Ular zamonaviy OT larning aksariyat ko'pchiligida asosiy hisoblanadi. FBT i disk sohasini markazlashgan holda taqsimlash va ma'lumotlarni boshqarish muammosini yechadi.

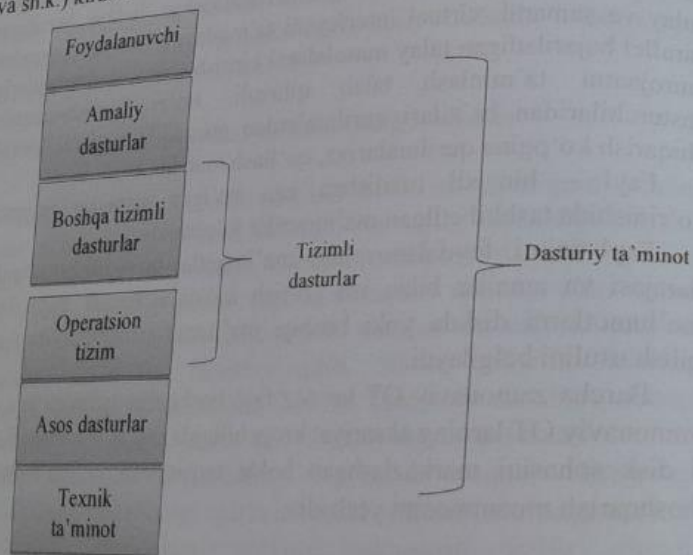
1.3. Hisoblash tizimining tarkibiy qismlari

Operatsion tizim haqida tushunchaga ega bo'lish uchun dastlab hisoblash tizimi umuman olganda nimalardan tashkil topganini ko'rib

chiqish lozim. Hisoblash tizimining umumiy strukturasi 1.1-rasmda tasvirlangan.

Birinchiidan, hisoblash tizimi (HT) texnik ta'minotdan iborat (inglizcha hardware): protsessorlar, xotira, monitorlar, disk qurilmalari, tarmoq kommunikatsion qurilmasi, printerlar va hokozo, bularning barchasi magistrat shinalar bilan bog'langan bo'ladi[2].

Ikkinchiidan, HT tizimli va amaliy qismlardan iborat bo'lgan dasturiy ta'minotdan (DT) iborat. Tizimli DT — bu HTning komponentalari bo'lmish protsessor, kommunikatsion va periferiya qurilmalarini boshqaradigan dasturlar to'plami bo'lib tizimni yaxlitligicha ishlashini ta'minlashga qaratilgan. Tizimli DT amaliy DTdan umuman farq qilib, foydalanuvchi muammolarini hal qilgan holda, ma'lum foydalanuvchi masalalarini bajarishga va foydalanuvchi bilan bevosita o'zaro aloqasiga qaratilgan. Amaliy DTlarga, odatda, turli yordamchi dasturlar (o'yinlar, matn muharrirlari, grafik dasturlar va sh.k.) kiradi.



1.1-rasm. Foydalanuvchi va hisoblash tizimining umumiy tuzilishi.

Yuqoridagilarni ma'lumotlardan kelib chiqadiki, operatsion tizim tizimli dasturiy ta'minotning fundamental komponentasi hisoblanadi. Dasturiy ta'minotning aynan shu qismini kengroq ko'rib chiqish predmeti bo'ladi.

Ma'lumki, operatsion tizim ixtiyoriy hisoblash tizimining asosiy komponentasidir va uning yaxlit holda ishlash samaradorligini belgilaydi. Shu bilan birga operatsion tizimga bitta ta'rif berish noo'rindir. Buning asosiy sababi, operatsion tizim bir qator turli funktsiyalarni bajaradi, ya'ni, foydalanuvchi-dasturchilarning hisoblash tizimining apparat qismi bilan qulay interfeysini ta'minlashdan tortib hisoblash tizimining resurslarini ratsional boshqarishni ta'minlashgacha vazifalarni o'z ichiga oladi. Shularni inobatga olgan holda bir nechta tushunchalarni keltirish mumkin.

Operatsion tizimlarni ishlab chiqaruvchilarning asosiy maqsadi quyidagilar:

1. Barcha kompyuter resurslaridan samarali foydalanish.
2. Dasturchilarning mehnat samaradorligini oshirish.
3. Hisoblash jarayonlarini tashkil qilishning soddaligi, samaraliligi va qulayligi.
4. Amaliy dasturiy ta'minotning apparat dasturiy ta'minotga bo'g'liq bo'lmasligini ta'minlash.

Hisoblash tizimi tarkibi konfiguratsiya deb ataladi. Odatda hisoblash texnikasining apparat va dasturiy vositalari alohida olib o'rganiladi. Shuning uchun ham mos ravishda hisoblash tizimlari apparat konfiguratsiyasi va dasturiy konfiguratsiyasi alohida olib o'rganiladi. Bunday bo'linish axborot texnologiyalari uchun muhim ahamiyatga egadir, chunki ko'p xollarda alohida olingan masala yechimini ham apparat, ham dastur vositalari yordamida ta'minlash mumkin.

Apparat ta'minoti. Hisoblash tizimlarining apparat ta'minoti tarkibiga, apparat konfiguratsiyani tashkil etuvchi qurilma va asboblardan iborat. Zamonaviy kompyuter va hisoblash majmua (kompleks)lari blok-modulli konstruktsiya (tuzilish)dan iborat. Ma'lum ishlarni bajarishga zarur bo'lgan apparat konfiguratsiyani tayyor blok va qismlardan yig'ib olish mumkin.

Qurilmalarning, markaziy protsessorga (Central Processing UNIT, CPU) nisbatan joylashishiga qarab tashqi va ichki qurilmalarga ajratamiz.

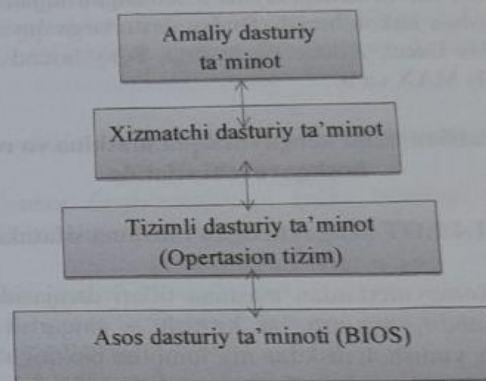
Tashqi qurilmalar, qoida bo'yicha, ma'lumotlarni kiritish va chiqarish qurilmalaridir, ularni odatda periferik qurilmalar ham deb ataladi. Bundan tashqari ma'lumotlarni uzoq saqlashga mo'ljallangan qurilmalar ham tashqi qurilmalarga kiradi.

Alohida blok va qismlar orasidagi kelishuvchanlik, birgalikda ishlashdagi moslanuvchanlik, apparatli interfeys deb ataluvchi o'tish apparat-mantiqiy qurilmalari yordamida bajariladi. Hisoblash texnikasidagi apparat interfeysiga belgilangan standartlar protokollar deyiladi. Shunday qilib, protokol – bu qurilma yaratuvchilari tomonidan, bu qurilmaning boshqa qurilmalar bilan muvaffaqiyatli va kelishilgan holda birgalikda ishlashi uchun, ishlab chiqiladigan texnik shartlar majmuasidir.

Dasturiy ta'minot. Dastur – buyruqlarning tartiblangan ketma-ketligidir. Kompyuterda, dasturiy va apparat ta'minot, doimo uzilmas aloqada va uzluksiz bog'lanishda ishlaydi. Biz bu ikki kategoriyani alohida ko'rib chiqayotganimizga qaramasdan, ular orasida dialektik aloqa mavjudligi va ularni alohida ko'rib chiqish shartli ekanligini esdan chiqarmaslik kerak. Dasturlar orasida xuddi kompyuterning fizik qismlari orasidagi kabi o'zaro aloqa mavjud. Aksariyat ko'pgina dasturlar, quyiroq darajadagi boshqa dasturlarga tayanib ishlaydi. Bunday bog'lanish dasturlararo interfeys deyiladi. Bunday interfeys (muloqot) ning mavjudligi texnik shartlar va o'zaro aloqa qoidalariga asoslangan bo'lsa ham, amalda u dasturiy ta'minotni o'zaro aloqada bo'lgan bir nechta sathlar (daraja)larga taqsimlash bilan ta'minlanadi.

Dastur ta'minoti sathlari piramida tuzilishiga egadir (2-rasm). Har bir keyingi sath oldingi sathlar dasturiy ta'minotiga tayanadi. Bunday ajratish, hisoblash tizimining dasturlarni o'rnatishdan boshlab, to amalda ekspluatatsiya qilish va texnik xizmat ko'rsatishgacha bo'lgan ish faoliyatining hamma bosqichlari uchun qulaydir. SHunga alohida etibor berish kerakki, har bir yuqoridagi sath butun tizimning funktsionalligini oshiradi. Masalan, asos dasturiy ta'minoti sathiga ega bo'lgan hisoblash tizimi ko'p funktsiyalarni bajara olmaydi, ammo u tizimli dasturiy ta'minotni o'rnatishga imkon beradi, ya'ni sharoit yaratadi.

Asos dasturiy ta'minoti – dasturiy ta'minotning eng quyi sathi hisoblanadi. Bu sathdagi dasturiy ta'minot apparat vositalari bilan o'zaro aloqaga javob beradi. Odatda, asos dasturiy ta'minoti bevosita asos qurilmalari tarkibiga kiradi va doimiy xotira deb ataladigan maxsus mikrosxemalarda saqlanadi. Doimiy xotiraga dasturlar kompyuterni ishlab chiqarish paytida yoziladi, ishlash jarayonida undagi ma'lumotlarni oddiy foydalanuvchilar o'zgartira olmaydilar faqat shu yo'nalishdagi dasturchilar unga o'zgartirish kiritadilar.



1.2-rasm. Dasturiy ta'minot sathlari.

Xizmatchi dasturiy ta'minotlar (utilitalar) bir vaqtning o'zida asos va tizimli dasturiy ta'minot bilan bog'langan bo'ladi. Bu sathdagi dasturlarning vazifasi kompyuter tizimini tekshirish, sozlash va tuzatishdan iborat. Odatda bunday dasturlar tizimli dasturiy ta'minot funktsiyalarini kengaytirishga va soddalashtirishga qaratilgan bo'ladi.

Xizmatchi dasturlarga quyidagilar kiradi:

- Fayl menejerlari;
- Arxivatorlar;
- Xujjatlarni ko'zdan kechirish vositalari;
- Diagnostika qiluvchi vositalar;
- Monitoring vositalari;
- O'rnatuvchi vositalar;

- Kommunikatsiya vositalari;
- Xavfsizlik vositalari.

Amaliy dasturiy ta'minotini o'z navbatida 2 guruhiga bo'lish

mumkin: dasturlash tizimlari va ilovalar.
Dasturlash tizimlari dasturchi uchun dasturlash vositasidir, bular: Visual Basic, Delphi, Python va b. Ilovalar foydalanuvchiga dasturlashni bilmay turib, matnli, grafik, sonli, audio va videoaxborotlarni qayta ishlashga, multimedia texnologiyalaridan foydalanishga, shuningdek kompyuter tarmoqlarida ishlashga imkon beradi. Ilovalar har xil turdagi fayllar hisoblangan hujjatlarni yaratishga va o'zgartirishga imkon beradi. Buday dasturlarga quyidagilar kiradi: Ms Word, Ms Excel, Adobe Photoshop, Sony Sound Forge, Adobe Premiere, 3Ds MAX va b.

1.4. Operatsion tizim kengaytirilgan mashina va resurslarni boshqaruvchi sifatida

1.4.1. OT kengaytirilgan mashina sifatida

Ko'pgina kompyuterlardan mashina tillari darajasida foydalanish ancha murakkabdir, ayniqsa bu kiritish – chiqarish masalalariga tegishlidir. M-n, yumshoh diskdan ma'lumotlar blokini o'hishni tashkil etish uchun dasturchiga 16 turli komandalardan foydalanishiga to'hri keladi, ularning har biri 13 ta parametrlarni aniqlashni talab qiladi, ya'ni masalan: diskdan blok tartib raqami, yo'ldagi sektor tartib raqami va h.k.lar. disk bilan bajariladigan amal tugallanishi bilan, kontroller, taxlil qilinishi kerak bo'lgan xatolik mavjudligini va tiplarini ko'rsatuvchi 23 ta qiymatni qaytaradi. Kiritish va chiqarish masalalarini dasturlashni real xaqiqiy muammolariga chuqur e'tibor bermagan holda ham, dasturchilar orasida bu amallarni dasturlash bilan shug'ullanishni hohlovchilar topilishi dargumondir. Disk bilan ishlashda dasturchi – foydalanuvchiga, diskni har biri o'z nomiga ega bo'lgan fayllar to'plamidan iborat deb tasavvur qilish kifoyadir.

Fayl bilan ishlash, uni ochish, o'qish va yozish amallarini bajarish va faylni yopishdan iboratdir. M-n, bunda, chastotali modulyatsiyani mukammallashtirish yoki o'hiydigan mexanizm "golovka" lari holati, joyini o'zgarish kabi savollar foydalanuvchinibezovta qilishi kerak

emas. Dasturchidan qurilmalar (apparatura) mohiyatini hammasini yashirib, unga ko'rsatilgan fayllarni qulay va sodda o'qish, yoki yozish, ko'rishni imkonini beradigan dastur – bu albatta OT dir. Xuddi shu kabi, OT dasturchilarni disk jamlamasi apparaturasidan ajratib, unga oddiy fayl interfeysini taqdim etadi, va bu holda OT uzilishlarni qayta ishlash, taymerni va operativ xotirani boshqarish va talay shu kabi quyi darajadagi muammolar bilan bog'liq yoqimsiz amallarni o'z zimmasiga oladi. Har bir holda, foydalanuvchi, real apparatura bilan ish ko'rish o'rniga muloqot uchun qulay va soddadir. Bu nuqtai nazardan, OT foydalanuvchiga ma'lum kengaytirilgan yoki virtual mashinani taqdim etadiki, uni dasturlash ham oson va u bilan ishlash soddadir, albatta bu real mashina takshil etadigan apparatura bilan bevosita ishlash qulay va yengildir.

1.4.2. Operatsion tizim resurslarni boshqaruvchi sifatida

OT, avvalambor foydalanuvchiga qulay interfeys yaratuvchi degan ho'ya albatta, masalani yuqoridan pastga qarab nazar solishga mos keladi.

Boshqa nuqtai nazar, ya'ni pastdan yuqoriga qarab nazar tashlash, bu OT ga murakkab tizimning hamma qismlarini boshqaruvchi mexanizm nazar solishdir.

Zamonaviy hisoblash tizimlari, protsessorlar, xotira, taymerlar, disklar, jamharmalar, Tarmoq kommunikatsiyaqurilmalari, printerlar va boshqa qurilmalardan iboratdir. Ikkinchi yondashishga mos ravishda OT ning funktsiyasi, protsessorlar, ya'ni resurslarni rahobatdosh jarayonlar orasida taqsimlashdan iboratdir. OT hisoblash mashina resurslarini jamisini shunday boshqarish kerakki, uni ishlashi maksimal samaradorlikni ta'minlashi zarurdir. Samaradorlik ko'rsatkichi, m-n, tizim o'tkazuvchanlik hobilyati yoki reaktivligi bshlishi mumkin.

Resurslarni boshqarish, masala resursi tipiga bog'liq bo'lmagan ikkita umumiy masalani yechishni o'z ichiga oladi:

- resursni rejalashtirish - ya'ni berilgan resursni kimga, hachon va taqsimlashdan iboratdir;

- resurs holatini kuzatish – resursni band yoki bo'shligi, bo'linadigan resurslar hahida esa resursning qancha qismi esa

taqsimlanmaganligi haqidagi operativ ma'lumotni olib turishdan iboratdir.

Resurslarni boshqarishni umumiy masalasini yechishda, turli OT lar turli algoritmlardan iboratdir, bu esa o'z navbatida OT larni umumiy qiyofasi, unumdorlik xarakteristikalari, qo'llanilish sohalari va hatto foydalanuvchi interfeysini yuqori darajada OT vaqtni bo'lish tizimi, paketli ishlov berish tizimi yoki real vaqt tizimiga mutanosibligini belgilaydi.

1.5. Nazorat savollari

1. Operatsion tizimning vazifasi nimalardan iborat?
2. OT resurslarni boshqarish tizimi sifatida deganda nimani tushunasiz?
3. OT foydalanuvchi va dasturlar himoyachisi deganda nimani tushunasiz?
4. OT virtual mashina sifatida tushunchasiga ta'rif bering?
- 5.1-chi davr kompyuterlarida qanday dasturlardan foydalanishgan?
- 6.2-chi davr kompyuterlarida qanday dasturlardan foydalanishgan?
- 7.3-chi davr kompyuterlarida qanday dasturlardan foydalanishgan?
- 8.4-chi davr kompyuterlarida qanday dasturlardan foydalanishgan?
- 9.5-chi davr kompyuterlarida qanday dasturlardan foydalanishgan?
10. Hisoblash tizimining texnik ta'minoti deganda nimani tushunasiz?
11. Hisoblash tizimining dasturiy ta'minoti deganda nimani tushunasiz?
12. Operatsion tizimlarni ishlab chiqaruvchilarning asosiy maqsadlarini sanab bering?
13. Dastur deb nimaga aytiladi?
14. Dasturiy ta'minot sathlarini sanab o'ting?
15. Asos dasturiy ta'minot vazifalari nimalardan iborat?
16. Xizmatchi dasturiy ta'minot vazifalari nimalardan iborat?

17. Qanday dasturlarga amaliy dasturlar deyiladi?
18. Operatsion tizim kengaytirilgan mashina sifatida.

2. OPERATSION TIZIM QURISH TAMOYILLARI

2.1. OT ni qurish asosiy printsiplari

2.1.1. Chastota printsipti

Dastur algoritmlarida, ishlov beriladigan massivlarda amal va kattaliklarni foydalanish chastotasiga qarab ajratishga asoslangan. Ko'p marta ishlatiladigan amal va ma'lumotlarga tezroq murojaat qilishni ta'minlash uchun, ularni operativ xotiraga joylashtiriladi. Bunday murojaatning asosiy vositasi, ko'p sathli rejalashtirishni tashkil etishdir. Uzoq muddatli rejalashtirishga tizim faoliyatining kamyob va uzun amallari ajratilsa, qisqa muddatli rejalashtirishga esa ko'p ishlatiladigan amallari ajratiladi. Tizim dasturlash bajarilishini initsializatsiya va qisqa amallar ajratiladi. Tizim dasturlash bajarilishini initsializatsiya qiladi yoki uzadi, dinamik tarzda talab qilinadigan resurslarni beradi va qaytib oladi, eng birinchi navbatda bu resurslar – xotira va protsessordir.

2.1.2. Modullilik printsipti

Modul-bu tizimning tugallangan elementi bo'lib, u modullararo interfeysga mos ravishda bajarilgandir. Modul ta'rifi bo'yicha, uni ixtiyoriy boshqasiga, mos interfeys mavjud bo'lganda almashtirish imkonini nazarda tutadi. Ko'pincha, OTni qurishda imtiyozga ega bo'lgan, qayta kiradigan va rinterabel modullar katta ahamiyatga egadir. Imtiyozga ega bo'lgan modullar imtiyozli rejimda amalga oshadi, bu rejimda uzilishlar tizimi o'chiriladi, va xech qanday tashqi xodisa hisoblashlar ketma-ketligini buza olmaydi. reinterabl modullar bajarilishni (ijroni) ko'p marta, takroran uzilishini va boshqa masalalardan qayta ishga tushirishni nazarda tutadi. Buning uchun, oraliq hisoblashlarni saqlash va uzilgan nuqtadan ularga qaytish ta'minlanadi. qayta kiradigan modullar ko'p marta parallel foydalanishni nazarda tutadi, ammo uzilishni nazarda tutmaydi. Ular imtiyozli bloklardan tashkil topgan bo'lib, ularga qayta murojaat, bu bloklarning birortasining tugallanganidan keyin mumkin bo'ladi. Modullilik printsipti, tizimning texnologik va ekspluatatsiya xossalarini aks ettiradi. Foydalanishning maksimal samaradorligi, agar bu printsipti OT ga ham, amaliy dasturlarga ham apparaturaga ham xos bo'lsa.

2.1.3. Funktsional tanlanish printsipti

Bu printsipti, hisoblashlar unumdorligini oshirish maqsadida, doimiy ravishda operativ xotirada bo'lishi kerak bo'lgan modullarni ajratishni nazarda tutadi. OT ning bu qismi yadro deyiladi.

Bir tomonda operativ xotirada qancha modullar ko'p bo'lsa, amallar bajarilish tezligi shuncha yuqori bo'ladi. Boshqa tomondan, yadro band qiladigan xotira xajmi juda katta bo'lishi mumkin emas, chunki aks holda amaliy masalalarga ishlov berish samarasi past bo'ladi. Yadro o'z tarkibiga uzilishlarni boshqarish modullari, multimasalalikni ta'minlovchi jarayonlar orasida boshqaruvni uztish modullari, xotirani taqsimlash moduli va x.k.larni oladi.

2.1.4. OT ni generatsiya qilish printsipti

Bu printsipti, yechiladigan masala va hisoblash tizimining konfiguratsiyasidan kelib chiqqan holda, OT ni sozlashga imkon beradigan OT yadrosi arxitekturasini tashkil etish printsiptini belgilaydi. Bu protsedura juda kam hollarda, OT ni uzoq vaqt davomida ekspluatatsiya qilish oldidan bajariladi.

Generatsiya jarayoni maxsus generator-dasturi va mos kirish tili yordamida amalga oshiriladi. Generatsiya natijasida OT ning, tizimli modul va kattaliklardan iborat to'liq versiyasi vujudga keladi. Modullilik printsipti generatsiyani ahamiyatli darajada soddalashtiradi. Bu printsipti ayniqsa Linux OT larida yaqqol ko'zga tashlanadi, unda nafaqat OT yadrosi generatsiya qilinadi, yuklanadigan tranzit modullari tarkibini ko'rsatadi. Boshqa OT larda konfiguratsiya qilish installatsiya jarayonida bajariladi.

2.1.5. Funktsional ortiqchalilik printsipti

Bu printsipti aynan bir amalni, har xil vositalar bilan bajarish imkoniyatini hisobga oladi. OT tarkibiga resurslarni boshqaruvchi bir necha xil monitorlar va fayllarni boshqaruvchi bir necha xil monitorlar va fayllarni boshqaruvchi bir nechta tizimlar va x.k.lar kiradi. Bu esa o'z navbatida, OT ni hisoblash tizimini aniq konfiguratsiyasiga tez va yetarli darajada moslashishga, aniq sinf masalalarini yechishda texnik

vositalarni samarali yuklashni maksimal ta'minlashga va shunda maksimal unumdorlikka erishishga olib keladi.

2.1.6. Standart holatlar printsipti (po umolchaniyu)

Tizim bilan ishlashda, ham generatsiya bosqichida ham, tizimlar bilan bog'lanishni tashkil etishni engillashtirish uchun qo'llaniladi. Printsip tizimidagi foydalanuvchi dasturini xarakterlovchi va ularning bajarilish vaqtini oldindan aniqlovchi, qurilma konfiguratsiyasi, modullar va jarayonlar strukturasi tavsiflarini tizimda saqlashga asoslangandir. Bu ma'lumotni foydalanuvchi tizimi, ma'lumot berilmagan bo'lsa, yoki atayodan aniqlashtirilmagan bo'lsa, foydalanadi. Umuman, bu printsipni qo'llash, foydalanuvchi tizim bilan ishlayotgan vaqtda, u o'rnatadigan parametrlarni qisqartirish imkonini beradi.

2.1.7. Joyini o'zgartirish printsipti

Bu printsip modullarning bajarilishi, ularning xotirada joylashgan o'rniga bog'liqligini ko'zda tutadi. Modul matnini, uni xotirada joylashuviga mos ravishda sozlash maxsus mexanizmlar, yoki uning bajarilishi davomida amalga oshiriladi. Sozlash, komandalarning adres qismida foydalanadigan haqiqiy adreslarni aniqlashdan iborat bo'lib, ayni OTlar uchun qabul qilingan operativ xotirani taqsimlash algoritmi va qo'llaniladigan adreslash usuli bilan aniqlanadi. U foydalanuvchi dasturlariga ham taqsimlanadi.

2.1.8. Virtuallashtirish printsipti

Bu tizim yagona markazlashgan sxemadan foydalanib, tizim strukturasi, jarayonlarni rejalashtiruvchilar (planirovshiklar) va resurs (monitorlari) taqsimlovchilari ma'lum majmuasi ko'rinishida tasvirlashga imkon beradi. Virtuallik kontseptsiyasi, virtual mashina tushunchasida akslanadi. Ixtiyoriy OT, haqiqatda, foydalanuvchidan, real apparat va boshqa resurslarni yashirib, ularni ma'lum abstraksiyalar bilan almashtiradi. Natijada, foydalanuvchilar virtual mashinani, ularning dasturlarini qabul qiluvchi va ularni bajarib, natija

beruvchi etarli darajadagi abstrakt qurilma sifatida foydalanadilar va tasavvur qiladilar. Foydalanuvchini, umuman hisoblash tizimi real konfiguratsiyasi va uning komponentalaridan samarali foydalanish qiziqitirmaydi. Bir nechta parallel jarayonlar uchun, bir vaqtning o'zida real tizimda mavjud bo'lmagan narsadan bir vaqtda foydalanish tasavvuri hosil qilinadi.

VM, real arxitekturani ham aks ettirishi mumkin, ammo bu holda arxitektura elementlari ko'pincha sistema bilan ishlashni soddalashtiruvchi, mukammallashtiruvchi yangi parametrlar bilan chiqadilar. Foydalanuvchi nuqtai-nazarida, ideal mashina quyidagilarga ega bo'lishi kerak:

- ishlashi mantihi jixatidan bir xil tarzda, chegaralanmagan xajmga ega bo'lgan virtual xotira;
- parallel ravishda bir-biriga ta'sir qiladigan va ishlay oladigan virtual protsessornlarning ixtiyoriy mihdori;
- virtual mashina xotirasiga ketma-ket va parallel, sinxron va asinxron murojaat etishga hodir bo'lgan virtual Tashqi qurilmalarning ixtiyoriy mihdori (soni) ma'lumotlar xajmi chegaralanmaganda ideal mashinaga yahinlashtirilgan, OT tomonidan amalga oshiriladigan virtual mashina qanchalik katta bo'lsa, ya'ni arxitekturali mantiqiy xarakteristikasi realdan qanchalik farq hilsa, demak virtuallikning shunchalik yuqori darajasiga erishilgan bo'ladi. OT bir-biri ichiga joylashtirilgan VM ierarxiyasi sifatida huriladi. Dasturlarning quyi sathi mashinaning apparat vositalaridir.

Keyingi sath esa dasturiy bo'lib, quyi sath bilan birgalikda, mashina yangi xossalarga ega bo'lishiga yordam beradi. har bir yangi sath ma'lumotlarga ishlov berish funktsiya imkoniyatlarini kengaytirish imkonini berib, quyi sathlarga murojaatni osonlashtiradi.

VM larni ierarxik tartibga solish ustunliklarga ega bo'lish, ya'ni loyixa doimiyliigi, dastur tizimlari ishonchliligi, ishlab chiqish muddatlari hisharishi, qator muammolarga ega. Ularning asosiylari: virtuallashtirish sathlari sonini va hossalarni aniqlash, OT ning har bir sathiga zaruriy qismlarni kiritish qoidalarini aniqlash.

Abstraktlashtirish (virtualizatsiya) alohida sathlari xossalari:

1. Har bir sathda, yuqori sathlar mavjudligi va xossalari to'g'risida hech narsa ma'lum emas.

2. Har bir sathda, boshqa sathlar ichki tuzilishi to'g'risida xech narsa ma'lum emas. Ular orasidagi bog'lanish oldindan belgilangan hat'iy qoidalar orhali olib boriladi.

3. Har bir sath bir nechta moduldan iborat, ularning ba'zilari ichki hisoblanadi va ularga boshqa sathlar murojaat qilishi mumkin. holgan modullar nomi yuqori sathlarga ma'lum va shu sathlar bilan bohlana oladi.

4. Har bir sath ma'lum resurslarga ega, u o'z resurslari abstraksiyalarini (virtual resurslarni) boshqa sathlardan yashirishi yoki taklif qilishi mumkin.

5. Har bir sath, tizimda ma'lumotlarning ma'lum abstraksiyasini ta'minlaydi.

6. Har bir sathda, boshqa sathga nisbatan qilinayapgan taklif minimal bo'lishi shart.

7. Sathlar orasidagi bog'lanish aniq argumentlar, bir sathdan ikkinchisiga uzatiladigan argumentlar bilan chegaralangan bo'lishi kerak.

8. Global ma'lumotlardan bir nechta sathlar foydalanishi mumkin emas.

9. Har bir sath boshqa sathlar bilan mustaxkamroh va kuchsiz bog'lanishi kerak.

10. Abstraksiya sathi orqali bajariladigan har qanday funktsiya yagona kirishga ega bo'lishi kerak.

2.1.9. Dasturiy ta'minotni Tashqi qurilmalarga bog'liq emasligi (muustahilligi) printsipt

Bu printsipt, dasturning aniq qurilmalar bilan bog'lanishi, dastrularni translyatsiya darajasida emas, balki undan foydalanishni rejalashtirish davridaligidan iboratdir. Dasturlarning yangi qurilmalar bilan ishlashi vaqtida, qayta kompilyatsiya qilinishi talab qilinmaydi. Bu printsipt ko'pgina OTlarda amalga oshiriladi.

2.1.10. Mutanosiblik printsipt (sovmestimost)

Bu printsipt, bir OT uchun yaratilgan dastur ta'minotining (DT) boshqa OT va shu OT ning oldingi versiyalarida ham bajarilish

imkoniyatini belgilaydi. Mutanosiblik ijro fayllari va dastur berilgan matni darajasida bo'lishi mumkin. Birinchi holatda tayyor dasturni boshqa OTda ishga tushirish mumkin. Buning uchun mikroprotessor komandasi, tizimli va kutubxona chaqirihlari darajasidagi mutanosiblik talab qilinadi. qoida bo'yicha, mashina kodini qayta kodlash imkonini beradigan va ularni boshqa protsessorlar terminlaridagi ekvivalent komandalar ketma-ketligiga almashtiradigan maxsus ishlab chiqiladigan emulyatorlardan foydalaniladi. Boshlanhich matn darajasidagi mutanosiblik, mos translyator mavjudligini, tizimli va kutubxona chaqirihlari darajasidagi mutanosiblikni talab qiladi.

2.1.11. Ochiqlik va qo'shimcha imkoniyatlar qo'shish printsipt

Ochiqlik taxlil uchun nafaqat tizimli mutaxassislarga balki foydalanuvchilarga ham imkoniyat borligini ko'zda tutadi. qo'shimcha imkoniyatlar qo'shish, OT tarkibiga yangi modullar qo'shish va mavjudlarini o'zgartirish (modifikatsiya) imkonini beradi.

OTni mikroyadro strukturasiidan foydalanib, klient-server texnologiyasiga asosan qurish, qo'shimcha imkoniyatlar qo'shish keng imkoniyat yaratadi. Bu xolda OT imtiyozli boshqaruvchi dasturlar va imtiyozsiz server-xizmatlar majmuasi tarzida huriladi. Asosiy qism o'zgartirilmasdan qolib, serverlar oson o'zgartiriladi, almashtiriladi va qo'shimcha qo'shiladi.

2.1.12. Mobillilik printsipt (ko'chirib o'tkazish)

Bu printsipt OTni bir platformadan, boshqa tipdagi platformaga ko'chirish imkonini nazarda tutadi. Ko'chirib o'tkaziladigan OT ni ishlab chiqishda quyidagi qoidalarga rioya qilinadi: OTning deyarli katta qismi, foydalanishga mo'ljallangan hamma platformalarda translyatorlari mavjud bo'lgan tilda yoziladi. Bu yuqori darajadagi, qoida bo'yicha S tilidir. Assemblerdagi dastur umumiy holda, ko'chirib bo'lmaydigan dasturdir. Keyin, apparat resurslari bilan bevosita munosabatda bo'lgan kod fragmentlari olib tashlanadi yoki kamaytiriladi. Apparatga bog'liq kod, bir nechta yaxshi lokallashtirilgan modullarda ajratilgan holda bo'ladi.

2.1.13. Xavfsizlik printsiplari

Bir foydalanuvchi resurslarini boshqa foydalanuvchidan himoyani, va hamma tizimni resurslarni faqat bitta foydalanuvchi egallab olishidan himoyani ko'zda tutadi. Bundan tashqari bu printsiplari o'z ichiga, xuquqsiz murojaatdan himoyani ham oladi. NCSC (National Computer Security Center), 1981y chiqarilgan "oranjivaya kniga"ga asosan, tizimlar 7 ta kategoriyaga: D, C1, C2, B1, B2, B3, A1 ga bo'linadi, bu erda A maksimal ximoyalangan tizimdir. Aksariyat ko'pgina zamonaviy OT lar S2 sinfga mansubdir. Bu sinf quyidagilarni ta'minlaydi.

- foydalanuvchini yagona nom va parol bilan tizimga kirishga imkon beradigan, maxfiy kirish vositalari.
- resurs egasiga, uning resursidan foydalanishga kimning xuquqi bor-yo'qligini aniqlaydigan murojaatni tanlab nazorat qilish;
- hisobga olish va kuzatish (audit) vositalari, ular tizimli resurslarga murojaat va tizim xavfsizligi bilan bog'liq bo'lgan xodisalarni aniqlash va topishga imkonni ta'minlaydi. A sinfi tizimni, ma'lum xavfsizlik ko'rsatkichlariga formal, matematik jixatdan mosligini isbotlashini talab qiladi. A sinfida, xavfsizlikni boshqarish mexanizmi, protsessor vaqtining 90% ini band qiladi. OT da himoyani ta'minlashni bir nechta yo'nalishi amalga oshiriladi. Ulardan biri, protsessor ishini ikki kontekstda olib borish, ya'ni vaqtning har bir onida protsessor OT tarkibidagi dasturni yoki Ot tarkibiga kirmaydigan amaliy yoki xizmatchi dasturni bajarish mumkin. har qanday bo'linadigan resurslarga foydalanuvchi va xizmatchi dasturlar tomonidan bevosita murojaatni ta'hirlash uchun, mashina kodlari tarkibiga, resurslarni taqsimlovchi va foydalanishni boshqaruvchi maxsus imtiyozli komandalar kiritiladi. Bu komandalarni faqat OT ga bajarish ruxsat etiladi. Ularning bajarilish nazorati apparat qismi tomonidan bajariladi. Bunday komandani bajarishga xarakter qilingan holda uzilish ro'y beradi, va protsessor imtiyozli rejimga o'tkaziladi. himoya printsiplari amalga oshirish uchun, operativ xotiradagi dastur matni va ma'lumotlarni himoya qilish mexanizmidan foydalaniladi. Bunda eng ko'p tarhalgan usul-kontekst himoyalansiqidir. Dasturlar va foydalanuvchilar uchun xotiraning ma'lum qismi ajratiladi, va bu chegaradan chiqilsa himoya bo'yicha uzilish ro'y beradi. Nazorat

mexanizmi, apparat tarzda, registrlar chegaralanganligi va xotira kalitlari asosida amalga oshiriladi. Fayllarda ma'lumotlarni saqlashning har xil himoya usullari qo'llaniladi. Eng oddiy himoya usuli-parolli usuldir.

2.2. Nazorat savollari

1. OT ni qurish asosiy printsiplari.
2. Chastota printsiplari
3. Modullilik printsiplari.
4. Funktsional tanlanish printsiplari.
5. OT ni generatsiya qilish printsiplari.
6. Funktsional ortiqchalilik printsiplari.
7. Standart holatlar printsiplari (po umolchaniyu).
8. Joyini o'zgartirish printsiplari.
9. Virtuallashtirish printsiplari.
10. Dasturiy ta'minotni Tashqi qurilmalarga bog'liq emasligi (muustahilligi) printsiplari.
11. Mutanosiblik printsiplari (sovmestimost).
12. Ochiqlik va qo'shimcha imkoniyatlar qo'shish printsiplari.
13. Mobillilik printsiplari (ko'chirib o'tkazish).
14. Xavfsizlik printsiplari.

3. OPERATSION TIZIMNING ARXITEKTURASI

3.1. Operatsion tizim arxitekturasi (Yadro, buyruqlar protsessori, kiritish – chiqarish tizimi, fayl tizimi)

Birinchi dasturlar bevosita mashina kodlarida yaratilgan. Buning uchun, mikroprotsessor arxitekturasini va uning asosidagi tizimni mukammal bilish zarur. hisoblash texnikasining rivojlanishi borasida, ko'p uchraydigan amallarni (operatsiyalarni) ajratib, ular uchun dasturiy modullar yaratilib, keyinchalik ulardan dasturiy ta'minotda foydalana boshladilar. Shunday qilib, 50-chi yillarda, birinchi dasturlash tizimlarini yaratishda kiritish chiqarish amallari uchun, keyinchalik matematik amal va funktsiyalarni hisoblash uchun, modullar yaratildi. Keyinchalik, rivojlanish yuqori darajadagi translyatorlarni yaratilishiga olib keldi, ya'ni operatorlar o'rniga zaruriy funktsiyalarni chaqiriqlarni qo'yish imkoniyati kelib chiqdi. Kutubxonalar soni oshib bordi. Natijada, amaliy dasturiy ta'minot ishlab chiquvchilaridan tizim arxitekturasini mukammal bilish talab qilinmay qoldi. Ular, dasturiy tizimga mos chaqiriqlar bilan murojaat va ulardan kerakli servis va funktsiyalarni olish imkoniga ega bo'ldilar. Bu dasturiy tizim OT dir.

Zamonaviy OT asosiy tashkil etuvchilari – bu yadro, kiritish-chiqarish tizimi, komanda protsessori, fayl tizimi. Yadro masalalar va resurslarni boshqarish, sinxronlashtirish va o'zaro bog'lanishi bo'yicha asos funktsiyalarni ta'minlaydi. Komanda protsessori, komandalarni qabul qilish va ularga ishlov berish, foydalanuvchi talabi bo'yicha mos xizmatlarni chaqirishni ta'minlaydi.

Kiritish va chiqarish tizimi, Tashqi qurilmalar bilan ma'lumotlarni kiritish va chiqarish masalasini ta'minlaydi.

OT kutubxonalarida bu funktsiyalarning mavjudligi, har bir ularni dasturlash tizimi vositalari bilan har bir dasturga qo'shmaslik imkonini beradi. Dasturlash tizimlari faqat, kiritish-chiqarish tizimi kodlariga murojaatni generatsiya qiladi va kattaliklarni tayyorlaydi. KCh tizimi, kiritish-chiqarish qurilmalari turlari ko'p bo'lgani uchun eng murakkab hisoblanadi. Bunda, nafaqat samarali boshqarish, balki amaliy dasturchilarga qurilmalardan abstraktlashtirishga imkon beradigan qulay va samarali virtual interfeysni ta'minlaydi. Boshqa tomondan,

parallel bajariladigan talay masalalarni kiritish-chiqarish qurilmalariga murojaatni ta'minlash talab qilinadi. Ba'zi KCh dasturchilaridan ba'zilar qurilmalardan mustahildir, ularni KCh ko'pgina qurilmalariga, qo'llash mumkin.

Fayl – bir xil tuzilishga ega bo'lgan yozuvlar majmuasi ko'rinishida tashkil etilgan ma'lumotlar to'plamidir.

Fayl tizimi, foydalanuvchiga ma'lumotlar tuzilishining mantiqiy darajasi va amallar bilan ish ko'rish imkonini beradi. Fayl tizimi, ma'lumotlarni diskda yoki boshqa ma'lumot jamlanmasida tashkil qilish usulini belgilaydi.

Barcha zamonaviy OT lar o'z fayl boshqaruv tizimiga ega. Ular zamonaviy OT larning aksariyat ko'pchiligida asosiy hisoblanadi. FBT i disk sohasini markazlashgan holda taqsimlash va ma'lumotlarni boshqarish muammosini yechadi.

Zamonaviy universal operatsion tizim birinchidan fayl tizimlarini ishlatadigan va foydalanadigan ko'p foydalanuvchili va ko'p vazifali xarakteristika deb aytishimiz mumkin.

Operatsion tizimning tarkibida asosiy 3 komponent guruhi mavjud:

1. Yadrosi (rejalashtiruvchi tizim, qurilma drayverlari, qurilmalarni boshqaradigan xizmatlar, tarmoq tizimlari)
2. Tizim kutubxonalari (drayverlar bilan birga keladigan vazifali lug'at)
3. Platalar bilan obolochkasi (tashqi ko'rinish)

Protssorni boshqarish algoritmi bo'yicha operatsion tizimlarning turlanishi:

1. Bir vazifali va ko'p vazifali
2. Bir foydalanuvchili va ko'p foydalanuvchili
3. Bir protssorli va ko'p protssorli tizim
4. Lokal va tarmoqlar

Bir vaqt davomida bajariladigan vazifalar bo'yicha operatsion tizimlar 2 klassga bo'linadi: bir vazifali va ko'p vazifali

Misol: MS – DOS operatsion tizim 1 vazifali, Linux, Unix, Windows ko'p vazifali hisoblanadi.

Ko'p vazifali operatsion tizimlar foydalanadigan muhitiga nisbatan 3 turga bo'linadi:

1. Paketlarni qayta ishlash tizimlari
2. Vaqt bo'yicha ajralish tizimlari

3. Real vaqt tizimlari
 Bir vaqt davomida ishlab turgan foydalanuvchi bo'yicha turlanishi:
 Bir protsessorli va ko'p protsessorli tizimlar.
 Operatsion tizim afzalliklaridan biri bir protsessor hamda ko'p protsessorni ma'lumotlarini qayta ishlashini ta'minlashidir. Hisob – kitob qilish harakatlar tashkillashtirish bo'yicha ushbu operatorlar simmetrik va assimetrik bo'lishi mumkin. Lokal va tarmoq operatsion tizimlar; lokal operatsion tizimlar avtonom kompyuter tizimlarida qo'llanilishi mumkin.

Operatsion tizimning funksiyali vazifalari:
 1. Lokal resurslarni boshqarish
 2. Jarayonlarni boshqarish
 Jarayon (vazifa, harakat) – abstrakt amal bajarib turgan dasturni aniqlaydi.

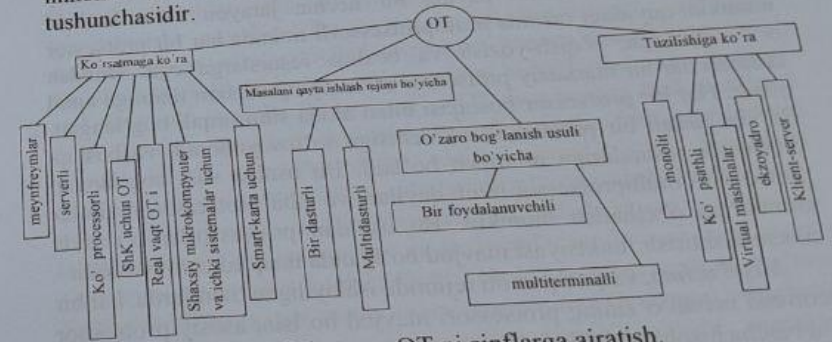
1. Kiritish – chiqarishni boshqarish
 2. Fayl tizimi har bir ma'lumot ustida ishlashni foydalanuvchi uchun qulay interfeysni ta'minlashi shart. Undan tashqari o'zaro bir – biri bilan aloqasi yo'q jarayonlarni yoki foydalanuvchilarni bir faylga murojaat qilishini ta'minlab berishi kerak.

Operatsion tizimning funksiyalari:
 1. Foydalanuvchi tomonidan buyruq va vazifalarni qabul qilishi va ularni qayta ishlashi
 2. So'rov dasturni qabul qilish va bajarish kerak bo'lgan vaqtda boshqa dasturlarni bajarish yoki to'xtatish
 3. Bajarishga mo'ljallangan dasturlarni operativ xotiraga yuklash
 4. Dasturni initsializatsiyasi
 5. Dasturlarni identifikatsiyasi
 6. Ma'lumot bazalarining fayllarini boshqarish tizimini ishlashini ta'minlab berish

7. Multidasturlash rejimini ta'minlab berish
 8. Hamma kiritish – chiqarish operatsiyasini boshqarish va tashkillashtirishni ta'minlash
 9. Real vaqt davomida vaqtni cheklashi bilan aniq va qattiq ishlash
 10. Xotirani taqsimlash
 a. Virtual xotirani tashkillashtirish
 b. Zamonaviy tizimlarda doimiy xotirani taqsimlash
 11. Vazifa bo'yicha rejalashtirish va nazorat qilishni ta'minlash

12. Bir vaqt mobaynida bajarib turgan dasturlar orasida ma'lumot va xabarlarini almashishini ta'minlab berish
 13. Bir dasturni boshqa dastur ta'siridan himoyalash va ma'lumotlarni ishonchli ravishda saqlash
 14. Har vaqtda tizimni ishdan chiqish vaziyatlari uchun xizmatlarni ta'minlash
 15. Dastur tizimining ishlashini ta'minlash

FBT, foydalanuvchilardan kiritish-chiqarish qurilmalari va disk xususiyatlarini yashirgan holda, fayl va kataloglar bilan ishlash uchun keng xizmat (servis) imkoniyatlarini beradi. UNIX da disklarda fayl tizimi va asos fayl tizimi alohida-alohida mavjuddir va ular bir-biriga bog'liq emas. UNIX da disk nomi yoki uning nomerini qo'shish bog'lanishni keltirib chiqaradi. "Mount" tizimi chaqirig'i yumshoq disk tizimini asos fayl tizimidagi ko'rsatilgan joyga qo'shish (montirovka) imkonini beradi. UNIX da yana bir tushuncha- maxsus fayllar tushunchasidir.



3.1-rasm. OT ni sinflarga ajratish.

OT tizimli boshqaruvchi va ishlov beruvchi dastur majmuasidan iborat bo'lib, bular dastur modullari va ma'lumotlar tuzilmalarining o'zaro bog'langan murakkab tuzilmasidir, ular hisoblashlarning ishonchli va samarali bajarilishini ta'minlashi zarurdir. OT ning ko'pgina potentsial imkoniyatlari, uning texnik va iste'mol parametrlari – bularning hammasi, tizim arxitekturasi OT ning strukturasi, qurilish printsiplari bilan belgilanadi.

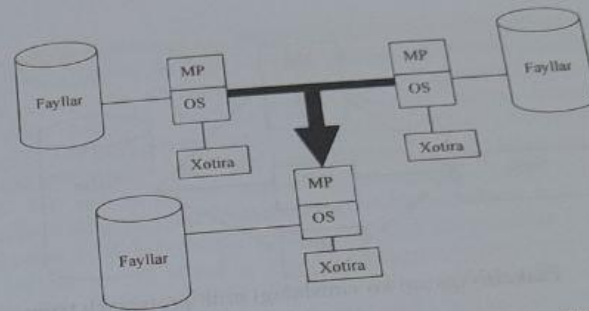
Shu vaqtgacha biz OT larga tashqi tarafdan nazar soldik, ya'ni OT qanday funktsiyalarni bajaradi. Endi esa, OT larni ichki qismini va ularni qurishda qanday yondashishlar mavjudligini ko'rib chiqamiz.

Zamonaviy OT larga qo'yiladigan talablarni qondirishda uning strukturali qurilishi (tuzilishi) katta ahamiyatga egadir, OT lar o'z rivojlanish bosqichlarida, monolit tizimlardan to yaxshi strukturalashtirilgan, rivojlanish xususiyatiga ega bo'lgan, kengaytirish va yangi platformalarga o'tkazuvchanlik xususiyatiga ega bo'lgan tizimlarga bo'lgan yo'lni bosib o'tdilar.

3.2. Ko'p protsessorli ishlov berish, paketli ishlov berish, real vaqt tizimlari, vaqt ajratish, taqsimlangan tizimlar

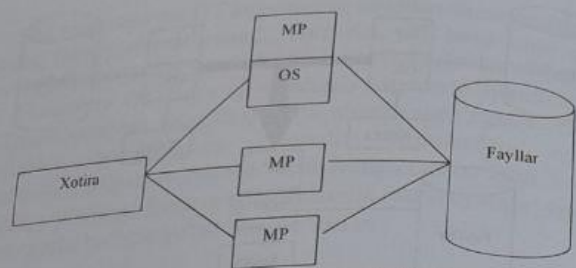
Ko'p protsessorli operatsion tizimli mashina bir protsessorli tizimlardek amal qilishi kerak. Farqi shundan iboratki, jarayonlar rejalashtiruvichisi foydalanuvchi topshiriqlari uchun bittadan markaziy protsessor ajratishi, bir paytda bir nechta jarayon aktiv bo'lishi mumkin. quyidagi rasmda multiprotsessorli tizimda har bir protsessor o'z xotirasiga, o'qish-yozish va boshqa resurslarga ega. Bundan tashqari har bir markaziy protsessor o'zining operatsion tizimiga amal qiladi. Har bir protsessor boshqasi bilan aloqa simi orqali bog'langan. Bu sim orqali bir protsessor ikkinchisiga so'rov yuboradi va boshqa protsessor resurslariga murojaat bo'ladi. Bu usulga sust bog'langan protsessorli multiprotsessor tizim deyiladi va u bir protsessorli tizimlar tormog'iga o'xshatish mumkin. Bu usuldan protsessorlar o'rtasida maxsuslashtirish funktsiyasi mavjud bo'lganda maqsadga muvofiqdir.

Misol uchun, vaqt taqsimoti rejimida ishlaydigan tizimlarda, har bir terminal uchun o'zining protsessori mavjud bo'lsin, asosiy protsesoor esa barcha hisoblash ishlarini bajaradi (zarur bo'lganda barcha terminal protsessorlari bilan bog'langan holda).



Protsessorlari sust bog'liq bo'lgan multiprotsessorli tizim

Sust bog'langan protsessorli multiprotsessor tizimlar nisbatan sodda, chunki har bir resur qandaydir bitta protsessorga ajratiladi. Ikkinchi tomondan, resurslar protsessorlar o'rtasida to'g'ridan-to'g'ri taqsimlanishi mumkin. Bu holga kuchli bog'langan protsessorli multiprotsessor tizimlar deyiladi va ular ma'lum ma'noda OT vazifasini qiyinlashtiradi. Multiprotsessorli tizimlarni tashkil qilishning yana biri etakchi-qaram usulidir. Resurslarni va OT ning boshqa funktsiyalarini bitta asosiy protsessor tomonidan boshqariladi. Protsessorlarning o'zaro aloqasi maxsus aloqa simlari orqali yoki birgalikda ishlatiladigan xotiraning ishchi sohalari orqali o'rnatiladi. Protsessorlar xotira va berilganlar, fayllar birgalikda ishlatilishi mumkin, lekin OT tashkil qiluvchi programmalari va berilganlar strukturasi faqat asosiy protsessor tomonidan foydalaniladi. OTning bunday tuzilishi ma'lum bir kamchiliklarga ega, asosiy protsessorlarning OT xizmati bilan band bo'lib qolganda boshqa protsessorlar "turib" qolishi mumkin, ikkinchidan asosiy protsessor ichida yuz berishi mumkin. Bu muammoni echish uchun har bir protsessorga OT va foydalanuvchi tomonidan talab qilinadigan funktsiyalarni bajarishga ruxsat berishdir.



Etakchi-qaram ko'rinishdagi multiprotsessorli tizim.

Bu usulga *simmetrik qayta ishlash tizimlar* deyiladi. Simmetrik tizimlarda OTning turli qismlari bir vaqtda amal qilish mumkin. Shu sababli bunday operatsion tizimni amalga oshirish boshqalarga nisbatan murakkab bo'ladi. Barcha protsessorlar OT tomonidan ishlatiladigan berilganlarga murojaat qila oladi. Bunda ikkita protsessorlar bitta resurslarni ishlatishda o'zaro kelisha olmasligi mumkin. Bu muammoni echish uchun OT resurslarga murojaat qilishning maxsus vositalaridan va kritik holat jadvalidan foydalanishi mumkin.

3.3. Ob'ektga yo'naltirilgan yondashish

Mikroyadro texnologiyasi, muntazam ravishda rivojlanadigan modulli tizimlarga asos bo'lgan bo'lsa ham, u to'liq ravishda tizimni kengaytirish imkonini ta'minlay olmadi. Hozirgi vaqtda, bu maqsadga eng to'g'ri keladigan ob'ektga mo'ljallangan yondashish, bu holda dasturni har bir komponentasi funktsiya jihatdan bir-biridan (boshqalaridan) ajratilganidir.

Bu yondashishning asosiy tushunchasi "ob'ekt"dir. Ob'ekt – dastur va ma'lumotlarning birligi bo'lib, u boshqa ob'ektlar bilan axborot uzatish va qabul qilish orqali bog'lanadi. Ob'ekt aniq buyum, amaliy dastur yoki xujjatni yoki ba'zi abstraktsiyalar – jarayon, hodisalarni aks ettirishi mumkin. Ob'ekt dasturlari funktsiyalari shu ob'ekt ma'lumotlari ustida bajariladigan amallar ro'yxatini belgilaydi. Ob'ekt – mijoz, ob'ekt – server funktsiyasini bajarilishini so'rab boshqa ob'ektga habar yuborishi mumkin.

Ob'ektlar mohiyatni tavsiflash mumkin. Umumiy ob'ektlardan aniqroq ob'ektlarni keltirib chiqaruvchi, meros berish mexanizmi xossalari mavjuddir. Ob'ekt ma'lumotlari ichki tuzilishi kuzatishdan yashirindir. Ob'ekt ma'lumotlarini ixtiyoriy tarzda o'zgartirib bo'lmaydi. Ob'ektdan ma'lumot olish yoki ma'lumotlarni ob'ektga joylashtirish uchun, mos ob'ekt funktsiyalarini chaqirish kerak bo'ladi. Bu hol ob'ektning, undan foydalanadigan koddan ajratadi. Dasturchi boshqa ob'ekt funktsiyalariga murojaat qilishi, yoki boshqa ob'ekt ular tuzilishini bilmay turib xossalarini meros qilish yo'li bilan yangi ob'ekt qurishi mumkin. Bu xossa inkapsulyatsiya deyiladi.

Shunday qilib, ob'ekt Tashqi dunyo uchun, yaxshi ishlab chiqilgan interfeysli "qora quti" ko'rinishida namoyon bo'ladi. Bu xususiyat, ularda mavjud ilovalarni ob'ekt ko'rinishida, ularda hech narsalarni o'zgartirmasdan, joylashtirishga imkon beradi.

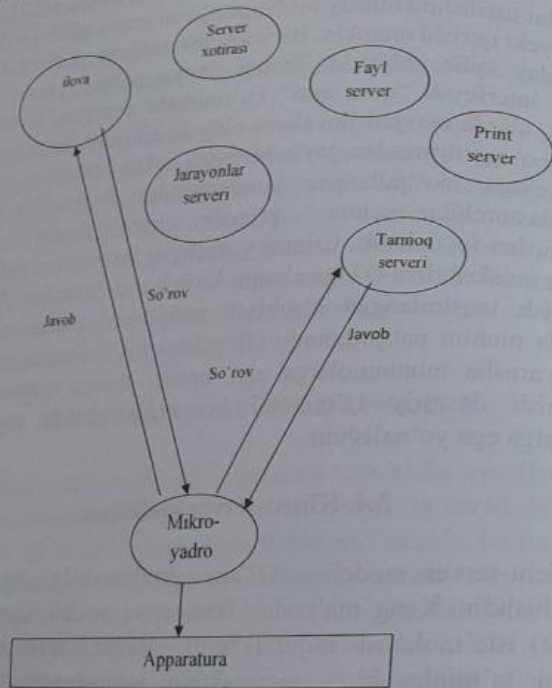
Ob'ektga mo'ljallangan yondashishdan foydalanish ayniqsa tizimli dasturchilar uchun qulaydir, chunki tizimli darajadagi ob'ektlardan foydalanib, tizimni yaxlitligini buzmasdan turib, ularni o'zlariga moslashtirib OT ga chuqur kirib borishi mumkin. Ammo bu yondashish taqsimlangan hisoblash muhitlarini amalga oshirishda kelajakda muhim natija beradi. Ob'ektga mo'ljallangan yondashish o'ziga yarasha muammolarga ega, ammo shunga qaramasdan, bu yondashish dasturiy ta'minotni konstruksiyalashda eng muhim ahamiyatga ega yo'nalishdir.

3.4. Klient-server modellari

Klient-server modeli – OT ni strukturalashtirishga yana bir yondashishdir. Keng ma'noda klient-server modeli, biror-bir servis (xizmat) iste'molchisi- mijoz (klient) – dastur komponentasi, va bu servisni ta'minlovchi – server-dastur komponentasi mavjudligini taqazo (nazarda tutadi) etadi.

Klient va server o'rtasidagi o'zaro aloqa shunday standartlashtiriladiki, server har xil usullar bilan amalga oshirilgan va, balki ishlab chiquvchi korxonalar turli bo'lgan klientlarga xizmat qilishi mumkin. Bunda asosiy shart, ular so'rovlarni serverga tushunarli usulda berishlari kerak.

Odatda ayirboshlash (obmen) sababchisi (initsiator) klient hisoblanadi, u kutish holatidagi serverga xizmat ko'rsatishni so'rab server bo'lishi mumkin. Bitta dastur komponenti ba'zi xizmatga nisbatan Bu model muvaffaqiyatli tarzda nafaqat OT larini qurishda, balki dasturiy ta'minotning barcha ko'rinishlarida (urovnyax) qo'llaniladi, faqat ba'zi hollarda tor-maxsus ma'noga egadir.



3.3-rasm. OT klient - server strukturasi.

OT ni strukturalashtirish g'oyasiga nisbatan olganda g'oya shundan iboratki, bunda uni bir nechta jarayonlarga-serverlarga bo'linadi, ularning har biri alohida servis funksiyalar majmuasini bajaradi – m-n, xotirani boshqarish, jarayonlarni yaratish yoki rejalashtirish. har bir server foydalanuvchi rejimida bajariladi. OT ni boshqa komponentasi

yoki amaliy dastur, servisni so'rab serverga xabar jo'natadi. OT yadrosi (mikroyadro) imtiyozli rejimda ishlagan holda, kerakli serverga xabar yetkazadi, server operatsiyani bajaradi, undan so'ng yadro, boshqa xabar bilan klientga natijani jo'natadi.

Bunda mikroyadrodan foydalanish, OT funksiyalarini vertikal taqsimlanishini gorizontaal taqsimlanishiga almashtirdi. Mikroyadrodan yuqorida joylashgan komponentalar, mikroyadro orqali yuboriladigan xabarlardan foydalansa ham bir-biri bilan bevosita aloqa qiladilar. Mikroyadro yo'naltiruvchi rolini o'ynaydi. U xabarni tekshiradi, serverlar va klientlarga jo'natadi va apparaturaga murojaat imkonini beradi.

Bu nazariy model klient-server tizimini mukammal tavsifdir, unda yadro faqat xabar uzatish vositalaridan iboratdir.

Haqiqatda esa, OT strukturasi amalga oshirilgan klient-server modeli turli variantlari, bir-biridan yadro rejimida bajaradigan ishlari xajmi bilan tubdan farq qilish mumkin.

Mikroyadro OT asosida yotadigan, eng muhim funksiyalarni amalga oshiradi. Bu nisbatan muhim bo'lmagan tizim xizmat va ilovalar uchun bazisdir.

Masala aynan shundan iboratki, tizimli funksiyalardan qaysi birini ahamiyatsiz deb hisoblab mos ravishda, yadroga qo'shmaslikdir, bu masala ko'p vaqtdan beri mikroyadro g'oyasi tarafdorlari orasida ko'pgina bahslarga sabab bo'lmoqda. Umumiy holda, OT ning an'anaviy tarzda ajralmas hisoblanadigan qismlari-fayl tizimlari, oynalarni boshqarish va xavfsizlikni ta'minlash – yadro bilan va o'zaro aloqa qiladigan periferik modullar bo'lib qolmoqda.

Bajariladigan ishlarni (funksiyalarni) mikroyadro va uni o'rab turgan modullar orasida taqsimlash asosiy prinsipi quyidagichadir: mikroyadro tarkibiga faqat supervizor rejimida va imtiyozli makonda ijro etiladigan funksiyalar kiritiladi. Bular mashinaga bog'liq funksiyalar (bir nechta protsessorni qo'llash funksiyasi ham), jarayonlarni boshqarish ba'zi funksiyalari, uzilishlarga ishlov berish, xabar jo'natishni qo'llash, kiritish-chiqarish qurilmalarini boshqarish ba'zi funksiyalari (qurilmalarni registriga komandalarni yuklash bilan bog'liq bo'lgan). Ot ning bu funksiyalarini foydalanuvchi makonida ishlayotgan dasturlarning bajarishi qiyin, ba'zi vaqtda esa mumkin emas.

Bu muammoni yechishning ikki yo'li bor. Birinchi yo'l, bir nechta protsessor ish rejimiga sezgir serverlarni yadro makoniga joylashtirishdir, bu esa o'z navbatida ularga apparaturaga to'liq murojaatni ta'minlaydi va shu bilan birga boshqa jarayonlar bilan oddiy usul bilan aloqani ta'minlaydi. Bunday yondashish, m-n, Windows NT ni ishlab chiqishda foydalanilgan: mikroyadrodan tashqari, imtiyozli rejimda, executive boshqaruvchi dastur deb ataluvchi qismi ishlaydi. U o'z ichiga virtual xotirani, ob'yektlarni, kiritish-chiqarish va fayl tizimlarini (Tarmoq drayvelarini) jarayonlar aloqasini va qisman xavfsizlik tizimini boshqaruvchi komponentalarni oladi.

Boshqa yo'l, yadroga, faqat serverning, yechimini amalga oshiradigan mexanizmdan iborat qismini qoldirishdan iborat, yechimni qabul qilishga javob beradigan qismini esa foydalanuvchi sohasiga joylashtiriladi. Qurilmalar drayverlari yadro tashqarisida ham, yadro ichida ham joylashgan bo'lishi mumkin. Qurilmalar drayverini yadrodan ajratish, OT dinamik konfiguratsiyasini mumkinligini imkonini beradi. Dinamik konfiguratsiyadan tashqari, qurilmalar drayverlarini foydalanuvchi rejimi jarayonlari sifatida qabul qilishning boshqa sabablari ham mavjud. MBBT, m-n, o'z drayveriga egadir. Bu yondashish, tizimni ko'chirib o'tkazish imkonini oshiradi, chunki ko'p hollarda qurilmalar drayverlari apparat qismidan abstraksiyalanadi.

Hozirgi vaqtda aynan klient – server modeli va mikroyadro kontseptsiyasidan foydalanib qurilgan OT lar, eng yuqori darajada, zamonaviy OT larga qo'yiladigan talablarga javob beradi.

Ko'chirib o'tkazishning yuqori darajasi, butun mashinaga bog'liq kod, mikroyadroga izolyatsiya (ajratiladi) qilinadi, shuning uchun ham tizimni yangi protsessorga ko'chirish kam o'zgarishlarni talab qiladi va ularning hammasi mantiqan birgalikda guruhlashtirilgan.

Mikroyadro texnologiyasi, ko'pgina turli OT lar uchun yozilgan dasturlar mutanosibligini ta'minlaydigan amaliy muhitlar asosidir. Quyida joylashgan OT dan, amaliy dasturlar interfeysini abstraksiyalashtirib, mikroyadro amaliy dasturlarga sarflangan mablag'lar bir necha yil davomida hatto OT va protsessorlar almashishi ham bekorga sarf bo'lmaydi.

Kengaytirishlik xossasi ham, zamonaviy OT larga qo'yilgan muhim talablardan biridir. OT DOS kabi kichik, yoki UNIX kabi

kattami, baribir ma'lum vaqtdan so'ng, uning konstruksiyasiga boshidan kiritilmagan xossalarga ega bo'lishi uchun, uni kengaytirish zarur bo'lib qoladi. Monolit OT larning o'sib borayotgan murakkabligi OT ga o'zgartirish kiritishni (uning keyinchalik ishi ishonchli bajariladigan holda) qiyinlashtiradi, ba'zi hollarda esa imkon ham bermaydi.

Mikroyadroning qat'iy belgilangan interfeysining chegaralangan OT faqat yadro rejimida bajariladi, amaliy dasturlar esa, ular yadroga tizimli funksiyalarni bajarish uchun yadroga murojaat qilgan hollardan boshqa hollarda foydalanuvchi rejimida bajariladi. Mikroyadroda qurilgan tizimlar, oddiy tizimdan farqli ravishda, o'zining server tizimlarining oddiy amaliy dasturlar kabi, foydalanuvchi rejimida bajariladi. Bunday tuzilish (strukturasi) mikroyadro yaxlitligiga ta'sir ko'rsatmasdan serverlarni o'zgartirish va qo'shish imkonini beradi.

server o'z xotira sohasida foydalanish ishonchlilikni oshiradi, har bir shunday qilib boshqa jarayonlardan himoyalangan. Agar alohida server buzilsa, u to'xtamasdan va OT ning boshqa qismlarini buzmasdan qayta ishga tushirish mumkin.

Bu model, taqsimlangan hisoblashlar uchun juda mosdir, chunki alohida serverlar multiprotsessorli kompyuterda yoki xatto har xil kompyuterlarda ishlashi mumkin.

3.5. Nazorat savollari

1. Operatsion tizim arxitekturasi
2. Ko'psathli tizimlar
3. Ko'p qatlamli (ko'p sathli) tizimlarda qatlamlar funksiyalarini aytib bering
4. Ko'p qatlamli tizimlar kamchiligi.
5. Klient-server modeli tuzilishi va ishlash printsiipi.
6. Mikroyadro modellarida modellar funksiyalarini aniqlash.
7. Ob'ektga yo'naltirilgan yondashishning o'ziga xos xususiyatlari.
8. Ko'p protsessorli ishlov berish
9. Simmetrik qayta ishlash tizimlar
10. Protsessorlari sust bog'liq bo'lgan multiprotsessorli tizim

11. Virtual mashinali va ko'p foydalanuvchili operatsion tizim
12. Imtiyozli rejim deb qanday rejimga aytiladi?
13. Mikroyadroli arxitekturaga asoslangan tizim deb qanday tizimga aytiladi?
14. Qanday tizimlarga ko'psathli tizimlar deyiladi?

4. OPERATSION TIZIMLARDA RESURS VA JARAYON TUSHUNCHASI

4.1. Operatsion tizimlarda jarayon va resurs tushunchasi

4.1.1. Operatsion tizimda jarayon tushunchasi

Operatsion tizimlarning ishini o'rganishda asosiy tushunchalardan biri bo'lib, ustida tizim aniq amallar bajaradigan asosiy dinamik ob'ektlar sifatida olinadigan jarayonlar hisoblanadi.

Oldingi ma'ruzalarda, OT tushunchasini qarab o'tganimizda, biz ko'pincha "dastur" va "topshiriq" so'zlarini ko'p ishlatdik. Masalan, xisoblash tizimi bitta yoki bir nechta dasturni bajaradi, OT topshiriqni rejalashtiradi, dasturlar ma'lumotlar almashadi va xokazo. Biz bu so'zlarni umumiy xolda ishlatdik va siz xar bir aniq xolda nimani tushunayapganingizni taxminan tasavvur qilar edik. Ammo bir xil so'zlar har xil xolatda masalan, xisoblash tizimi ishlov bermaydigan statik xolatdagi ob'ektlarni ham (masalan, diskdagi fayllar) va ijro jarayonidagi bo'lgan dinamik xolatdagi ob'ektlarni ham bildirar edi. Bu xolat OT lar umumiy xossalari to'g'risida gapirganimizda, ya'ni uning ichki qurilmalari va o'zini tutishiga e'tibor berilmagan xolatda mumkin edi. Ammo endi zamonaviy kompyuter tizimlari ishini detallashgan xolda o'rganganimizda biz amallarni (terminologiyani) aniqlashtirishimizga to'g'ri keladi.

"Dastur" va "topshiriq" terminallari statik, faol bo'lmagan (neaktivniy) ob'ektlarni tavsiflash uchun ishlatiladi. Dastur esa, bajarilish jarayonida dinamik, faol ob'ektga aylanadi. Uning ishlashi borasida kompyuter turli komandalarga ishlov beradi va o'zgaruvchilar qiymatlarini o'zgartiradi. Dastur bajarilishi uchun, OT ma'lum son operativ xotira ajratishi, unga kiritish chiqarish qurilmalarni va fayllarni bog'lashi, ya'ni butun hisoblash tizimi resurslari xisobidan ma'lum qismini rezervlab qo'yishi kerak. Ularning soni va konfiguratsiyasi vaqt o'tishi bilan o'zgarishi mumkin. Bunday kompyuter tizimlari ichidagi faol ob'ektlarni tavsiflash uchun "dastur" va "topshiriq" terminlari o'rninga yangi "jarayon" terminini ishlatamiz.

4.1.2. Operatsion tizimda resurs tushunchasi

Umumiy holda har iste'mol qilinadigan (uning), iste'molchilar uchun ma'lum qiymatga ega bo'lgan ob'ekt (uning), iste'molchilar zaxiradagi ajratiladigan birligiga qarab, **cheklangan resurs**dir. Resurslar bo'ladi. Resurs cheklanganligi iste'molchilar va **cheklanmagan** zaruriyatlarga olib keladi. Zaruriyatlarni tartibga solish uchun, hayotiy iste'molchilar orasida ularni yuqori darajada qoniqtiradigan ma'lum qoidalar bo'yicha taqsimlanishi zarur.

4.2. Resurslarni umumiy sinflarga ajratilishi

Resurslarning umumiy klassifikatsiyasini quyidagicha tasavvur qilish mumkin:

1. mavjudligi xaqiqiylikiga qarab: fizik va virtual
2. xossalarning kengaytirish imkoniga qarab: moslanuvchan (elastik) va qat'iy
3. faollik darajasiga qarab: faol va passiv
4. mavjudlik vaqtiga qarab: asosiy va ikkinchi darajali
5. tuzilishi (struktura) jihatidan: oddiy va murakkab
6. tiplanishi bo'yicha: ishlab chiqiladigan va iste'mol qilinadigan
7. amalga oshirish formasi bo'yicha: yumshoq va qattiq
8. funktsionallik ortiqchaligiga qarab: qimmat va arzon
9. foydalanish xarakteriga qarab: parallel foydalaniluvchi va ketma-ket foydalaniluvchi.

Mavjudligi xaqiqiylikiga qarab resurslar

Fizik resurslar deb, real mavjud bo'lgan va uni foydalanuvchilar orasida taqsimlanganda unga xos bo'lgan fizik xususiyatlarga ega bo'ladigan resursga aytiladi (saqlab qoladi).

Virtual resurs-fizik resursning qandaydir boshqacha modelidir. Virtual resurs, foydalanuvchi tasavvur qilgan holda mavjud emas. Model sifatida virtual resurs dasturiy-apparat formada amalga oshiriladi. Bu ma'noda virtual resurs mavjuddir. Ammo virtual resurs foydalanuvchiga u bilan ishlash vaqtida, nafaqat fizik resurs, ya'ni modellash tirish ob'ektiga xos xossalarnigina emas, balki unga xos bo'lmagan xossalarni ham namoyon qiladi.

Xossalarning kengaytirish imkoniga qarab resurslar
Xossalarni kengaytirish imkoniyati belgisi resursni qurish imkoniyatini xarakterlaydi. Virtuallashtirishga, ya'ni o'z xossalarni kengaytirish imkonini beradigan resurs-**moslanuvchan yoki elastik** deyiladi. **Qat'iy resurs**, bu o'z ichki xossalari bo'yicha virtuallashtirishga yo'l qo'ymaydigan resursga aytiladi.

Faollik darajasiga qarab resurslar

Faol (aktiv) resursdan foydalanishda, u boshqa resurs yoki jarayonlarni (yoki o'ziga nisbatan ham) o'zgartiruvchi amallarni qo'llaydi yoki o'zgartirishga olib keladi. **Passiv resurs** bunday xususiyatga ega emas. Bunday ob'ekt ustida mumkin bo'lgan amallar bajariladi, bunda uning holati o'zgaradi, ya'ni ichki yoki tashqi xarakteristikalarini. Markaziy protsessor – faol resurs, talab bo'yicha ajratiladigan xotira passiv resursdir.

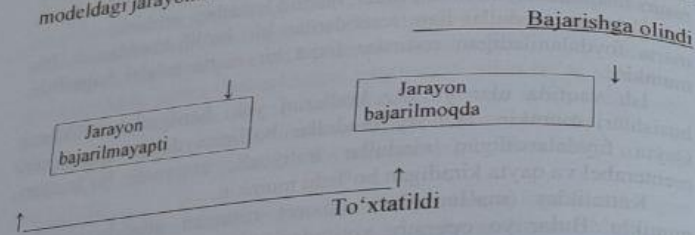
Mavjudlik vaqtiga qarab resurslar

“Mavjudlik (sushestvovanie) vaqti” belgisiga qarab farqlash, ulardan foydalanadigan jarayonlarga nisbatan olingan dinamikasiga qarab olib boriladi. Agar resurs tizimda jarayon tug'ilishigacha mavjud bo'lsa va jarayon mavjudligi intervalida murojaat qilish mumkin bo'lsa, u holda bu resurs ayni jarayon uchun doimiydir. Vaqtinchalik resurs ko'rilayotgan jarayon mavjudlik vaqtida tizimda dinamik holda paydo bo'lishi va yo'q qilinishi shu jarayon orqali yoki boshqa jarayonlar-tizimli yoki foydalanuvchi jarayonlari orqali olib borilishi mumkin. Demak, resurslar jarayonlarning tizimli o'zaro bog'lanish ma'lum qoidalari bo'yicha ajratiladi. SHuning uchun ham ba'zi jarayonlarga nisbatan doimiy bo'lgan resurs boshqasiga nisbatan vaqtinchalik yoki aksi bo'lishi mumkin.

Resurslarning muhimligi darajasiga nisbatan sinflarga ajratish zaruriyati, ikkita sabab bilan belgilangandir: zarur ishlovchanlik bilan ta'minlanishi va jarayonlarni boshqarish va resurslarni taqsimlashdagi moslanuvchanligini oshirish. Buning uchun asosiy va ikkinchi darajali resurslar ajratiladi. Resurs aniq jarayonga nisbatan asosiy bo'ladi, shu holdaki, agar uning taqsimlanishsiz jarayon rivojlana olmasa. Bunday resurslarga eng avvalo markaziy protsessor va operativ xotira kiradi. Agar resurslar taqsimlanmaganda jarayonni alternativ rivojlanishiga

Bir protsessorli kompyuter tizimida vaqtning har bir momentida faqat bitta jarayon bajarilishi mumkin. Multidasturli xisoblash tizimlarida bir nechta jarayonni psevdoparallel qayta ishlash protsessorni bir jarayondan ikkinchisiga o'tkazish yordamida amalga oshiriladi. Bir jarayon bajarilguncha, qolganlari o'z navbatini kutadi.

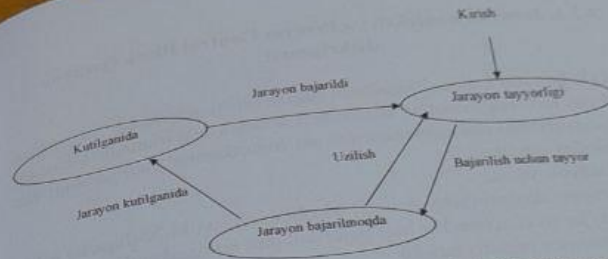
Ko'rinib turibdiki har bir jarayon minimum ikki xolatda bo'lishi mumkin: Jarayon bajarilmoqda va jarayon bajarilmayapti. Bunday modeldagi jarayonlar xolati diagrammasi quyidagi rasmda ko'rsatilgan.



Rasm 4.1. Jarayon holatini eng oddiy diagrammasi

Bajarilayapgan xolatdagi jarayon, ma'lum vaqtdan so'ng OT tomonidan tugallanishi, yoki to'xtatilishi va yana bajarilmayapgan xolatga o'tkazilishi mumkin. Jarayon to'xtatilishi ikkita sabab bilan ro'y berishi mumkin: uni ishini davom ettirishi uchun biror bir xodisa talab etilsa, (masalan, kirish - chiqish operatsiyasini tugallash) yoki OT tomonidan ish jarayon uchun ajratilgan vaqt tugaganda ro'y beradi. SHundan so'ng OT belgilangan algoritmi bo'yicha bajarilmayapgan xolatidagi jarayonlardan birini tanlaydi va bu jarayonni bajarilayapgan xolatga o'tkazadi. Tizimda paydo bo'layapgan yangi jarayon, boshqa jarayon bajarilmayapgan xolatga o'tkaziladi.

Bu juda qulay modeldir. U bajarilishga tanlangan jarayon u to'xtatilishiga sabab bo'lgan xodisani kutishi va amalda bajarilishga tayyor bo'lmasligi mumkin. Bunday xolatdan qutilish uchun jarayon bajarilmayapgan xolatni yangi ikkinchi xolatga bo'lamiz: tayyorlik va kutish xolatlari. (Rasm 4.2)



Rasm 4.2. Jarayon xolatining yanada to'liq diagrammasi

Tizimda paydo bo'layapgan xar qanday yangi jarayon tayyorlik xolatiga tushadi. Operatsion tizim rejalashtirishning biror bir algoritmidan foydalanib, tayyor jarayonlardan birini tanlab, uni bajarilish xolatiga o'tkazadi.

Bajarilish xolatida jarayon dasturiy kodini bevosita bajarilishi ro'y beradi. Jarayonning bu xolatidan uchta sabab bo'yicha chiqish mumkin: - OT bu jarayonning faoliyatini to'xtatadi;

- U o'z faoliyatini ma'lum xodisa ro'y bermaguncha davom ettira olmaydi va OT uni "kutish" xolatiga o'tkazadi;

- Xisoblash tizimida uzilish ro'y berishi bilan (masalan, bajarilishga ajratilgan vaqt tugashi bilan taymerdan uzilish) uni tayyorlik xolatiga o'tkaziladi.

Kutish vaqtlari tayyorgarlik xolatiga jarayon, kutilayapgan xodisa ro'y berishi bilan o'tadi va u yana bajarilish uchun tanlanishi mumkin. Keyinchalik rejalashtirish algoritmi xaqida so'z borsa, bizning modelda yana bir operatsiya qo'yiladi: bu jarayon prioritetini o'zgarishidir.

Jarayonni yaratish va tugallash operatsiyalari bir martalik operatsiyalardir, chunki ortiq qo'llanilmaydi ba'zi tizimli jarayonlar, xisoblash tizimi ishi vaqtida xech qachon tugallanmaydi.

Jarayon xolatini o'zgarishi bilan bog'liq bo'lgan, hox u ishga tushirish yoki blokirovka bo'lsin, qoida bo'yicha ko'p martalik xisoblanadi.

4.3.1. Jarayon konteksti va Process Control Block (jarayon diskriptori)

OT, jarayon ustidagi amallarni bajara olishi uchun, har bir jarayon OTda ma'lum ma'lumotlar strukturasi sifatida tasvirlanishi lozim bu struktura (tuzilma) shu jarayonga xos ma'lumotlarni o'z ichiga oladi. Bu ma'lumotlar quyidagilar:

- Jarayon xolati
 - Jarayon dasturli (schiyotchigi) hisoblagichi, yoki boshqacha qilib aytganda, jarayon uchun keyingi bajariladigan komanda adresi.
 - Protessor registri tarkibi.
 - Xotirani boshqarish va protsessordan foydalanishni rejalashtirish uchun zarur ma'lumotlar (jarayon prioriteti, adres makoni, o'lchami va joylashgan o'rni va xokazolar.)
 - Xisob (qayd) ma'lumotlari jarayon identifikatsiya nomeri, qaysi foydalanuvchi uning ishini initsializatsiya qildi, jarayonning protsessordan foydalanish umumiy vaqti va xokazolar
 - Kiritish -chiqarish qurilmalari bilan bog'liq ma'lumotlar (masalan, jarayonga qanday qurilmalar bog'langan, ochiq fayllar jadvali va xokazolar).
- Albatta bu ma'lumotlar tizimini va tarkibi xar bir OTga bog'liqdir. Ko'pgina OTlarda jarayonni xarakterlovchi ma'lumot bitta emas, balki bir nechta ma'lumotlar strukturasi saqlanadi. Bu strukturalar har xil nomlanishi, yuqorida keltirilgan ma'lumotlarni bir qismini yoki qo'shimcha ma'lumotlarni ham o'z ichiga olishi mumkin. Uni jarayon diskriptori, PCB (Process Control Block) yoki jarayonni boshqarish bloki deb nomlash mumkin.

4.3.2. Bir martalik amallar (operatsiyalar)

1. Jarayonni yaratish. 2. Jarayonni tugallatish

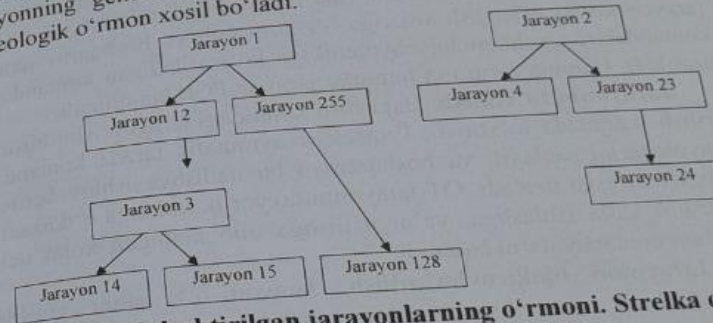
Jarayonning kompyuterdagi murakkab xayot yo'li uni tug'ilishidan boshlanadi. Jarayonlar kontseptsiyasini qo'llovchi ixtiyoriy OT, jarayon yaratish vositasiga ega bo'lishi kerak.

Eng oddiy tizimlarda (masalan, faqat bitta aniq ilova ishi uchun loyixalashtirilgan tizimlarda) xamma jarayonlar tizim satrida tug'ilishi

mumkin. Murakkabroq operatsion tizimlar, jarayonlarni zaruriyat bo'yicha dinamik xolda yaratadilar.

Operatsion tizim startidan so'ng, yangi jarayon tug'ilishi sababchisi bo'lib maxsus tizimli chaqiriq bajaragan foydalanuvchi jarayoni bo'lishi mumkin, ya'ni natijada yana jarayon yoki operatsion tizim bo'lishi mumkin.

YAngi jarayon tug'ilishiga sabab bo'lgan jarayon ota jarayon (parent process) deyiladi, qaytadan yangi yaratilgan jarayon - farzand jarayon deyiladi (child process). Farzand jarayon o'z navbatida yana yangi farzand jarayonni yaratishi mumkin va tizim ichida jarayonning geneologik daraxtini to'plamini xosil qiladilar, ya'ni geneologik o'rmon xosil bo'ladi.



Rasm.4.3 soddalashtirilgan jarayonlarning o'rmoni. Strelka ota - farzand munosabatini ko'rsatadi

Jarayon tug'ilishida tizim yangi PCB tuzadi bunda jarayon tug'ilishi xolati bilan uni to'ldira boshlaydi, yangi jarayon o'zining yagona identifikatsiya tartib raqamini oladi. Biror bir jarayon tugashi bilan bo'shagan identifikatsiya nomeri boshqa jarayon uchun ishlatilishi mumkin.

Odatda o'zining funksiyalarini bajarish uchun jarayon - farzand ma'lum resurslarni talab qiladi: xotira, fayllar, kiritish -chiqarish qurilmalari va xokazolar. Ularni ajratishning ikki xil xolati bor. YAngi jarayon o'ziga ota jarayon resurslarini olishi mumkin. Bunda u resurslarni ota -jarayon yoki boshqa farzand -jarayonlar bilan bo'lishadi yoki resurslarin bevosita operatsion tizimdan olishi mumkin. Ajratilgan resurslar xaqidagi ma'lumot PCBga kiritiladi.

4.3.3. Ko'p martalik operatsiyalar

Bir martalik operatsiyalar jarayonlar OT boshqaruvi ostidagi sonini o'zgarishiga olib keladi va har doim ma'lum resurslarni ajratilishi va bo'sh shashi bilan bog'liqdir. Ko'p martalik operatsiyalar operatsion tizimdagi jarayonlar sonini o'zgarishiga olib kelmaydi va resurslarni ajratish va bo'sh shashi bilan bog'liq bo'lishi shart emas.

Jarayonlar ustida ko'p martalik operatsiyalarni bajarish uchun OT qanday ishlarni bajarishini ko'rib chiqamiz.

Jarayonni ishga tushirish OT, tayyor turgan jarayonlar orasidan birini bajarishga tanlaydi. Tanlangan jarayon uchun uni bajarilishi uchun operativ xotirada zarur ma'lumot bilan ta'minlaydi. Keyin jarayon xolati bajarilish xolatiga o'zgartiriladi va boshqaruv jarayon komandalari xisoblagichi(schyotchik)ga ko'rsatiladigan komandasiga uzatiladi. Hamma zarur ma'lumotlar jarayon PCBdan olinadi.

Jarayonni to'xtatish. Bajarilish xolatidagi jarayon ishi, biror bir uzilish natijasida to'xtaydi. Protssessor avtomatik tarzda komandalar schyotchigini saqlaydi va boshqaruvni bu uzilishga ishlov beruvchi maxsus adresga uzatadi. OT jarayonni tayyorlik xolatiga o'tkazadi va uzilishni katta ishlashga, ya'ni uzilishga olib kelingan xolat uchun ma'lum operatsiyalarni bajaradi.

Jarayonni blokirovka qilish. Jarayon o'z ishini xisoblash tizimida biror bir xodisa ro'y bermaguncha davom ettira olmaydi. SHuning uchun jarayon ma'lum tizimli chaqiriq bilan OTga murojaat qiladi. OT tizimli chaqiriqni qayta ishlaydi(kiritish -chiqarish operatsiyalarini initsializatsiya qiladi, biror qurilmani bo'sh shashini yoki xodisa ro'y berishini kutayotgan jarayonlar navbatiga qo'shadi va xokazolar.), jarayonni bajarilish xolatidan kutish xolatiga o'tkazadi.

Jarayonni bloklashdan chiqarish(razblokirovanie). Tizimda biror xodisa ro'y bergandan so'ng, OT aynan qanday xodisa ro'y berganligini aniqlashi zarur. Keyin OT, qaysi jarayon shu xodisani kutish xolatida ekanligini aniqlaydi va shunday jarayon bo'lsa uni tayyorlik xolatiga o'tkazadi.(bunda OT xodisa ro'y berishi bilan bog'liq bo'lgan amallarni bajaradi.)

Protssessorni bir jarayondan ikkinchisiga to'g'ri(korrekt) o'tkazish uchun bajarilayotgan jarayon kontekstini saqlashi va protssessor o'tkazadigan jarayon kontekstini tiklash zarur. Bunday jarayonlar

ishlanganligini saqlash G'tiklash protsedurasi kontekstni o'tkazish deyiladi.

Jarayon tushunchasi. OT boshqaruvi ostidagi bajariladigan komandalar to'plami, ular bilan bog'liq resurslar va uning bajarilishi joriy momenti bilan xarakterlanadi. Ixtiyoriy vaqtida jarayon to'liq ravishda o'zining konteksti, ya'ni registirli, tizimli va foydalanuvchi qismlaridan tashkil topgan konteksti bilan tasvirlanadi. OTlarda jarayon aniq ma'lumotlar strukturasi -PCB bilan tasvirlanadi. PCB -registirli va tizimli kontekstlarni aks ettiradi. Jarayonlar beshta asosiy xolatlarda bo'lishi mumkin: tug'ilish, tayyorlik, bajarilish, kutish, bajarilishni tugallash.

Bir xolatdan ikkinchisiga jarayon OT yordamida, ular ustida biror bir amal bajarilishi natijasida o'tkaziladi. OT jarayonlar ustida quyidagi operatsiyalarni bajarishi mumkin: jarayon yaratish, jarayon tugallash, jarayonni to'xtatib turish, jarayonni blokirovka qilish, jarayonni bloklashdan chiqarish. Jarayon prioritetini o'zgartirish.

4.4. Jarayonlarni rejalashtirish algoritmlari va parametrlari

Xar gal, chegaralangan resurslar va ularning bir nechta iste'molchilari bilan ish ko'rilganda, masalan, misol uchun mexnat jamoasida maosh fondini taqsimlash deylik, biz mavjud resurslarni iste'molchilar o'rtasida taqsimlash bilan shug'ullanishimizga to'g'ri keladi, yoki boshqacha aytganda resurslardan foydalanishni rejalashtirishimizga to'g'ri keladi. Bunday rejalashtirish aniq qo'yilgan maqsadlarga (ya'ni, masalan, resurslarni taqsimlash xisobicha biz nimaga ega bo'lmoqchimiz) va bu maqsadlarga mos va iste'molchi parametrlariga tayanadigan algoritmlarga ega bo'lishi kerak.

4.4.1. Rejalashtirish ko'rsatkichi va algoritmlarga talablar

Jarayonlarni rejalashtirish xar bir darajasi uchun, xar turli juda ko'p algoritmlarni taklif qilish mumkin. Qaysi algoritmni tanlash, xisoblash tizimi ehadigan masalalar va biz rejalashtirishdan foydalanib erishmoqchi bo'lgan maqsadlarimizga bog'liqdir. Bu maqsadlar quyidagilardir.

• Xaqqoniylik –kompyuter tizimida, xar bir jarayon va topshiriq uchun protsessordan foydalanish vaqtining ma'lum qismi ajratilishiga kafolat berish. YA'ni, bir foydalanuvchi jarayonni xar doim protsessor vaqtini band qilishi va boshqa foydalanuvchi jarayoni bajarilmay turishiga yo'l qo'ymaslik.

• Samaradorlik –protsessor ish vaqtining xamma 100%ni band qilishga xarakat qilish. Bunda u, bajarishga tayyor jarayonlarni band turishi kerak emas. Real xisoblash tizimlarida protsessor yuklanishi 40ta 90%gacha o'zgarib turadi.

• To'liq foydalanish vaqtining qisqarishi(turn aroid time) – jarayonni starti yoki topshiriqni yuklashga navbat qo'yishi va uni tugallashi orasidagi minimal vaqtni ta'minlash.

• Kutish vaqtini qisqartirish() –jarayonlarning tayyor xolati va yuklashga navbatni berish vaqtini qisqartirish.

• Javob berish vaqtini qisqartirish –jarayonning interaktiv tizimlarda foydalanuvchi so'roviga javob berish uchun kerak vaqtini minimallashtirish.

Rejalashtirishning qo'yilgan maqsadlariga bog'liq bo'lmagan xolda, algoritmlar quyidagi xossalarga ega bo'lishi kerak.

• Aniq bo'lishi kerak, masalan, bitta topshiriq xar doim bir xil vaqda bajarilishi zarur.

• Minimal xarajatlar bilan bog'liq bo'lishi kerak. Masalan, protsessorning xar bir yuz millisekundiga, jarayon o'zining bajarilishiga qaysi protsessorni olishi mumkinligini aniqlash uchun 200 millisekund kerak bo'lsa, bunday algoritmni qo'llash maqsadga muvofiq emas.

• Xisoblash tizimi resurslarini bir xil taqsimlash zarur, bunda kam foydalaniladigan resurslarni band qiladigan jarayonlarga imtiyoz berish kerak.

• Masshtablashtirish xossasiga ega bo'lish, ya'ni yuklama oshganda ishlovchanlik qobiliyatini yo'qotmaslik.

Yuqorida keltirilgan maqsad va xossalalar bir –biriga qarama – qarshidir. Algoritmni bir kriterist(ko'rsatgich) nuqtasi nazaridan yaxshilasak, ikkinchisi nuqtai –nazaridan xolat yomon tomonga o'zgaradi.

4.4.2. Rejalashtirish parametrlari

Qo'yilgan maqsadlarni amalga oshirish uchun, yaxshi algoritmlar, tizimdagi jarayonlarning qandaydir xarakteristikalariga, yuklamaga navbatdagi topshiriqlarga xisoblash tizimi xolatiga, boshqacha qilib aytganda rejalashtirish parametrlariga tayanishi zarur.

Xamma rejalashtirish parametrlarini ikkita katta guruxlarga bo'lishi mumkin: statik parametrlar va dinamik parametrlar. Statik parametrlar xisoblash tizimi ish vaqtida o'zgaribmaydi, dinamiklari esa teskarisi, doimo o'zgarishda bo'ladi.

Tizimning statik parametrlariga uning resurslarining chegaraviy qiymatlarini (operativ xotira hajmi, svoping uchun diskdagi xotira maksimal soni, ulangan kiritish –chiqarish qurilmalarining soni va xokazolar). Tizimning dinamik parametrlari ayni vaqtdagi bo'sh resurslar sonini tavsiflaydi.

Jarayon statik parametrlariga, qoida bo'yicha yuklash vaqtiga xos xarakteristikalar kiradi.

• Jarayon qaysi foydalanuvchi tomonidan ishga tushirilgan va qaysi foydalanuvchi topshiriqni shakllantirgan.

• Qo'yilgan masala bajarilish prioriteti qanday, ya'ni masala qay darajada muxim.

• Foydalanuvchi tomonidan masalani echish uchun qancha protsessor vaqti so'ralgan.

• Protsessor va kiritish –chiqarish amalini bajarish vaqti nisbati qanday

• Topshiriq uchun, xisoblash tizimining qaysi rusurslari(operativ xotira, kiritish –chiqarish qurilmalari, maxsus kutubxonalar, tizimli dasturlar va xokazolar) va qancha miqdorda kerak.

Uzoq muddatga rejalashtirish algoritmlari o'z ishlarida xisoblash tizimining dinamik va statik parametrlaridan va jarayonlarning parametrlaridan (jarayonlar dinamik parametrlari topshiriqni yuklash etapida xali noma'lum bo'ladi).

O'rtacha muddatli va qisqa muddatli rejalashtirish algoritmlari, qo'shimcha ravishda jarayonlarning dinamik xarakteristikalaridan foydalanadilar. O'rtacha muddatli rejalashtirishda bunday xarakteristika sifatida quyidagi ma'lumotlardan foydalaniladi:

-Jarayonni diskka yoki operativ xotiraga yuklangan momentdan qancha vaqt o'tdi;

-Jarayon qancha operativ xotira egallaydi;

-Jarayonga qancha protsessor vaqti ajratildi;

Rejalashtirish jarayoni OTning "rejalashtiruvchi" deb ataladigan qismi orqali bajariladi. Rejalashtiruvchi, bajarishga, tayyor xolatdagi jarayon ichidan yangi jarayonni quyidagi to'rtta xollarda tanlash xaqida echim qabul qiladi:

1. Jarayon bajarilish xolatidan, bajarilish tugallandi xolatiga o'tishda

2. Jarayon bajarilish xolatidan kutish xolatiga o'tishda

3. Jarayon bajarilish xolatidan, tayyorlik xolatiga o'tishida

4. Jarayon kutish xolatidan, tayyorlik xolatiga o'tishida.

Rejalashtirishning turli-tuman algoritmlari mavjuddir, ular xar turli masalalar uchun samarali va turli maqsadlarga erishishga mo'ljallangandir.

Masalan, 1. First -come, First -Served (FCFS) -birinchi keldi, birinchi xizmat ko'rsatildi.

2. Round Robin (RR) -bolalar koruseli. Bu FCFS ni modifikatsiya qilingan ko'rinishidir.

3. Shortest - Job - First (SJF) -birinchining eng qisqa vaqti va xokazolar.

Hisoblash tizimi N ta foydalanuvchi interaktiv rejimda ishlayapgan bo'lsa, xar bir foydalanuvchi o'zida protsessor vaqtining -IG·N qismiga ega deb xisoblanishi kafolatlaydigan rejalashtirish algoritmini qo'llash mumkin.

Hisoblash tizimining eng chegaralangan resurslaridan biri protsessor vaqtidir. Ularni ko'p sonli jarayonlar orasida taqsimlash uchun tizimga jarayonlarni rejalashtirish protsedurasini qo'llashga to'g'ri keladi. Rejalashtirishning xisoblash tizimi xolatiga ta'sirining davomiyligi darajasiga qarab, jarayonlarni qisqa muddatli, o'rtacha muddatli va uzoq muddatli rejalashtirishlarga bo'linadi. Rejalashtirish aniq algoritmlari qo'yilgan maqsadlardan, echilayapgan masala sinflariga bog'liq bo'lib, jarayonlarning statik va dinamik parametrlariga va kompyuter tizimlariga tayanadi. Rejalashtirishning siqib chiqaradigan va siqib chiqarmaydigan rejimlari ajratiladi.

Siqib chiqarilmaydigan rejalashtirish rejimida, bajariladigan jarayon boshqa jarayonga protsessorni faqat xoxishi bilan berishi mumkin. siqib chiqaradigan rejimda esa, bajarilayapgan o'ziga bog'liq bo'lmagan xolda chiqariladi.

Eng oddiy siqib chiqarmaydigan rejalashtirish algoritmi -FCFSdir, u qisqa jarayonlarni sezilarli darajada ushlab qolishi mumkin (tayyorlik xolatiga vaqtida o'tmagan jarayonlarni).

Vaqtini ajratish tizimlarida keng tarqalgan algoritm bu siqib chiqaradigan algoritm -RRdir.

Siqib chiqaradigan algoritmlar ichida jarayonlarning o'rtacha kutish vaqti jixatidan optimal algoritm -SJF algoritmidir.

4.5. Semaforalar

Semaforalar jarayonlarni sinxronlashtirish uchun, aniqrog'i, bir nechta jarayonlarning umumiy resurslarga murojaatlarini sinxronlashtirish uchun ishlatiladi. IPC shakli sifatida semaforalar FIFO yoki xabarlar navbatida bo'lgani kabi katta hajmdagi ma'lumotlarni almashish uchun mo'ljallanmagan. Buning o'rniga, ular o'zlarining nomlariga to'liq mos keladigan vazifani bajaradilar - ya'ni ma'lum bir umumiy resursdan foydalanishga ruxsat berish yoki taqiqlash.

Semaforalardan foydalanishni oddiy misol bilan tushuntiramiz. Aytaylik, sizda bir xil umumiy resurs mavjud (masalan, fayl). Jarayon resursda operatsiyani amalga oshirganda (masalan, faylga yozish) boshqa jarayonlar uchun resursga kirishni blokirovka qilish kerak. Buning uchun biz ma'lum bir butun son qiymatini ushbu resurs bilan bog'laymiz - barcha jarayonlar uchun mavjud hisoblagich. Hisoblagichning 1 qiymati resursga murojaat qilish mumkinligini anglatadi, 0 murojaat mumkin emasligini anglatadi. Ushbu holatda, resurs bilan ishlashni boshlashdan oldin, jarayon hisoblagichining qiymatini tekshirishi kerak. Agar u 0 ga teng bo'lsa - resurs band va amal barjarish imkonsizdir. Agar hisoblagich qiymati 1 bo'lsa, siz resurs bilan ishlashingiz mumkin. Buning uchun, avvalo, resursni blokirovka qilish kerak, ya'ni hisoblagich qiymatini 0 ga o'zgartirish kerak, resursni bo'shatish uchun operatsiya bajarilgandan so'ng

hisoblagich qiymati 1 ga o'zgartirilishi kerak. Keltirilgan misolda hisoblagich semafor rolini o'ynaydi.

Normal holatda ishlash uchun quyidagi shartlar bajarilishini ta'minlash kerak:

- Semaforaning qiymati har xil jarayonlar uchun ruxsat etilgan bo'lishi kerak. Shuning uchun semafora jarayon manzili maydonida emas, balki yadro manzili maydonida joylashadi.
- Semaforaning qiymatini tekshirish va o'zgartirish operatsiyasi boshqa jarayonlarga nisbatan (ya'ni, boshqa jarayonlar bilan uzluksiz) bir atomar amal shaklida amalga oshirilishi kerak. Aks holda, semaforaning qiymatini tekshirgandan so'ng, jarayonning bajarilishini boshqa jarayon to'xtatib qo'yadigan holat yuzaga kelishi mumkin, bu esa o'z navbatida semafori tekshiradi va uning qiymatini o'zgartiradi. Amallarning muhim bo'limlari atomarligini kafolatlashning yagona usuli bu amallarni yadro rejimida bajarishdir.

Shunday qilib, semaforalar tizim murojaatlari interfeysi orqali amallar bajariladigan tizim resursidir.

Unixda semafori tavsiflash uchun `sem_t` yoziladi. Semafora `sem_init` funksiyasi yordamida yaratiladi. Misol:

```
#include <semaphore.h>
static sem_t mySemaphore;
...
int main() {
    ...
    sem_init(
        &mySemaphore, // semaphore
        0,             // shared between processes: No
        1              // G'G' initial value: semaphore is open
    );
    ...
}
```

Semafor obyektini ko'pincha statik xotirada global sifatida tavsiflanadi. Semafora yaratilganda uning boshlang'ich qiymati ko'rsatiladi (bu holda 1) va semaforaning turli jarayonlar o'rtasida bo'lishini belgilaydigan "bayroq" yaratiladi. Agar semaforadan turli xil jarayonlar foydalansa, u holda umumiy xotiraga joylashtirilishi kerak.

Foydalangandan so'ng, semaforani `sem_destroy` funksiyasi yordamida bo'shatish kerak:

```
sem_destroy(&mySemaphore);
sem_post funksiyasidan foydalangan holda semafora qiymatini bittaga oshiriladi:
sem_post(&mySemaphore);
Semaforani kutish sem_wait funksiyasi yordamida amalga oshiriladi:
sem_wait(&mySemaphore);
```

4.6. Nazorat savollari

1. OT da jarayon nima.
2. Jarayon holatlari va diagrammasi
3. Jarayon konteksti (ma'lumotlar)
4. Jarayon ustidagi amallar: bir martalik va ko'p martalik amallar.
5. Resurslardan foydalanishni rejalashtirish.
6. Rejalashtirish ko'rsatkichi va algoritmlarga bo'lgan talablar.
7. Resurslar ta'rifi va asosiy resurslarni aytib bering.
8. Resurslar qanday belgilar bo'yicha sinflarga ajratiladi.
9. Strukturaga ega resurslarni aytib o'ting.
10. Zamonaviy va birinchi OTlardagi qanday ob'ektlar resurslar deb hisoblangan?
11. Resurslarni virtuallashtirish deganda nimani tushunasiz.
12. Resurslarning qanday turlari mavjud.
13. mavjudligi xaqiqiylikiga qarab qanday turlari mavjud.
14. xossalarning kengaytirish imkoniga qarab qanday turlari mavjud.
15. faollik darajasiga qarab qanday turlari mavjud.
16. mavjudlik vaqtiga qarab qanday turlari mavjud.
17. tuzilishi (struktura) jihatidan qanday turlari mavjud.
18. tiplanishi bo'yicha qanday turlari mavjud.
19. amalga oshirish formasi bo'yicha qanday turlari mavjud.
20. funktsionallik ortiqchaligiga qarab qanday turlari mavjud.
21. foydalanish xarakteriga qarab qanday turlari mavjud.

5. OPERATSION TIZIMLARDA JARAYONLARNI BOSHQARISH

5.1. Operatsion tizimda jarayonlarni boshqarish

Rejalashtirish darajalari (urovni) quyidagilarga bo'linadi:

1. Uzoq muddatli.
2. O'rta muddatli.
3. Qisqa muddatli.

Yuqorida biz, xisoblash tizimidagi ikki xil rejalashtirish: topshiriqlarni va protsessordan foydalanishni rejalashtirish xaqida so'z yuritgan edik.

Topshiriqlarni rejalashtirish jarayonlarni uzoq muddatga rejalashtirish sifatida foydalaniladi. U, tizimda, uning multidasturlash darajasini, ya'ni bir vaqtning o'zida mavjud bo'lgan jarayonlar sonini aniqlab, jarayonlarni yuzaga kelishiga javob beradi. Agar tizimning multidasturlash darajasi doimiy bo'lib tursa, kompyuterdagi jarayonlar o'rtacha soni o'zgarmaydi, u holda yangi jarayonlar faqat oldin yuklanganlari tugallangandan keyin paydo bo'ladi. SHuning uchun xam uzoq muddatga rejalashtirish kam ishlatiladi chunki, yangi jarayonlar paydo bo'lishi orasida o'nlab minutlar o'tishi mumkin.

Protssordan foydalanishni rejalashtirish, jarayonlarni qisqa muddatga rejalashtirish sifatida foydalaniladi. U, masalan, bajariladigan jarayonning kiritish -chiqarish qurilmalari yoki vaqtning ma'lum intervali tugallangandan so'ng amalga oshiriladi. Masalan u, xam qisqa muddatli rejalashtirish 100 millisekundda bir marta kam amalga oshirilmaydi.

Ba'zi xisoblash tizimlarida, unumdorlikni oshirish uchun, qisman bajarilayapgan jarayonni operativ xotiradan diskka vaqtincha jo'natish va keyinroq esa uni bajarilishini davom ettirish uchun orqaga qaytarish mumkin. Bunday protsedura raswapping, ya'ni tarjimada "o'tkazish(perekachka)"ni bildirsa xam, tarjimasiz "svoning" termini ishlatiladi. Jarayonlardan qaysisini va qachon diskka va orqaga qayta o'tkazishni, odatda, jarayonlarni rejalashtirishning qo'shimcha darajasi -o'rtacha muddatli rejalashtirish yordamida amalga oshiriladi

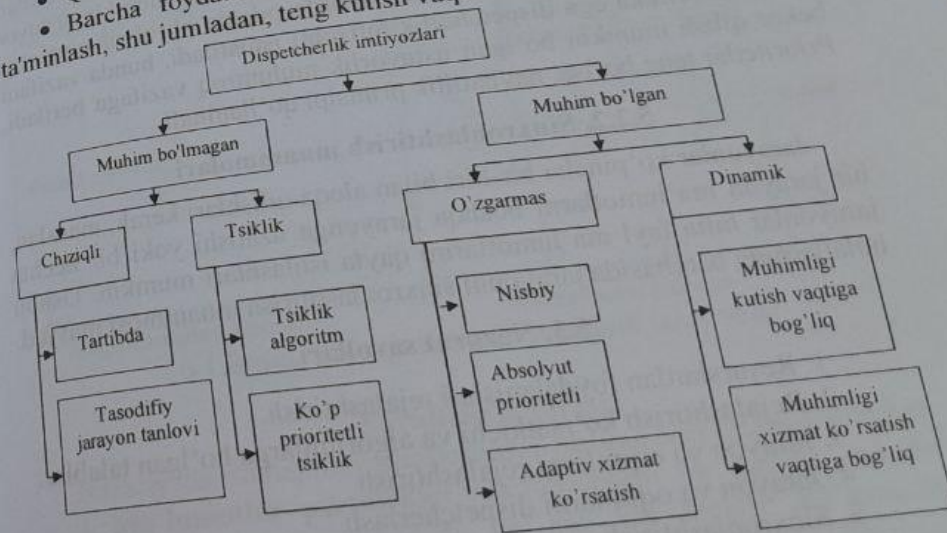
5.2. Rejalashtirish, dispetcherlash va sinxronizatsiyalash

5.2.1. Jarayon va oqimlarni rejalashtirish va dispetcherlash

Rejalashtirish strategiyasi uzoq muddatli va qisqa muddatiga bo'linadi. Uzoq muddatli rejalashtirish kontseptsiyasi har doim jarayonlarga taalluqlidir, qisqa muddatli rejalashtirish (dispetcherlash) esa - oqimlilarga tegishlidir. Agar OT oqimlilik qo'llab-quvvatlamasa, u holda jarayonlarni dispetcherlash amalga oshiriladi. Masalalar tushunchasini jarayonlar va oqimlar nomini umumlashtirish uchun ishlatiladi.

Uzoq muddatli rejalashtirish hisoblash tizimining resurslari uchun o'zaro kamroq raqobatlashadigan jarayonlarni to'plashdan iboratdir. Protssessor taqdim qilinishi lozim bo'lgan jarayonlarni tanlash strategiyasining eng ko'p uchraydiganlari:

- Hisoblash jarayonlarini ular qanday boshlangan bo'lsa shu tartibda tugatish;
- Qisqa jarayonlarga ustunlik bering;
- Barcha foydalanuvchi jarayonlarini bir xil xizmatlar bilan ta'minlash, shu jumladan, teng kutish vaqtini xam.



5.1 - rasm. Dispetcherlash intizomi

Ko'plab dispatcherlik intizomlari ma'lumki, ularga ko'ra bajarishga tayyor oqimlar navbati hosil bo'ladi. Ikkita katta dispatcherlik intizomlari mavjud - ustuvor bo'lmagan va ustuvor. Dispatcherlik intizomi ro'yxati 5.1-rasmda keltirilgan.

Eng oddiy dispatcherlik intizomi - bu o'z navbati bilan bajarilishi. Vazifalar paydo bo'lish tartibida bajariladi, ish jarayonida bloklangan vazifalar "tayyor" xolatiga o'tgandan so'ng xali bajarilmagan masalalar navbatining ohiriga o'tkaziladi. Bu "hisoblashlarni, iloji boricha, paydo bo'lish tartibida tugatish" xizmat ko'rsatish strategiyasini amalga oshirishdir. Hisoblash tizimining yuklanganlik darajasini oshishi bilan o'rtacha kutish vaqti xam ortib boradi, shu bilan birga qisqa vazifalar uzoq bajariladigan vazifalar bajarilguncha kutib turadi. Ushbu intizom paketli Operatsion tizimlarda qo'llaniladi. Vazifalarning davomiyligini inobatga oluvchi ushbu intizomning bir nechta modifikatsiyalari mavjud bo'lib, bunda birinchi navbatda qisqa vazifalar bajariladi.

Interfaol tizimlar uchun tizimning maqbul javob berish vaqtini ta'minlash maqsadga muvofiqdir agar tizim ko'p terminalli bo'lsa, xizmatlarning tengligi. Shuning uchun, hal qilish ushbu muammolardan tsiklik (karusel, dumaloq) intizom va foydalanish mumkin ustuvor xizmat usullari. Real operatsion tizimlarda odatda absolyut qiymatfani ega ustuvorlikka ega dispatcherlik intizomi ishlatiladi, bunda vazifani bekor qilish mumkin bo'lgan ustuvorlik muhimroq vazifaga beriladi. Prioritetlar teng bo'lsa, navbatlilik prinsipi qo'llaniladi.

5.2.2. Sinxronlashtirish muammolari

Jarayonlar ko'pincha bir-biri bilan aloqa qilishlari kerak, masalan, bir jarayon ma'lumotlarni boshqa jarayonga uzatishi yoki bir nechta jarayonlar bitta fayl ma'lumotlarini qayta ishlashlari mumkin. Ushbu holatlarning barchasida jarayonni sinxronlashtirish muammosi mavjud.

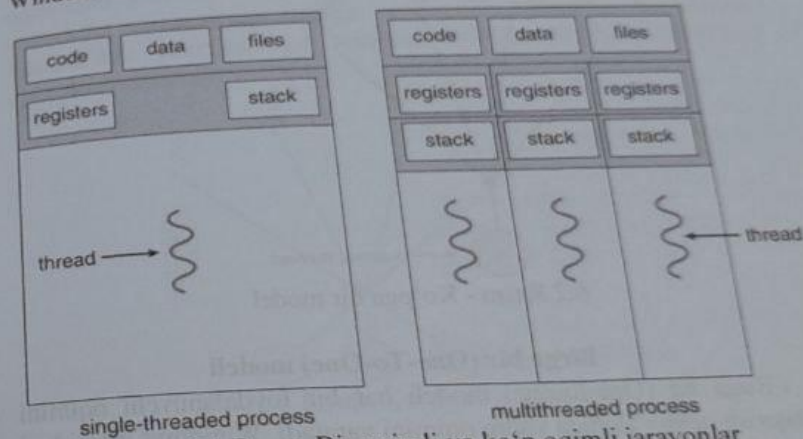
5.3. Nazorat savollari

1. Resurslardan foydalanishni rejalashtirish.
2. Rejalashtirish ko'rsatkichi va algoritmlarga bo'lgan talablar.
3. Jarayon va oqimlarni rejalashtirish
4. Jarayon va oqimlarni dispatcherlash
5. Sinxronlashtirish muammolari
6. Dispatcherlash intizomi

6. OPERATSION TIZIMLARDA OQIM(THREADS)LAR

6.1. Operatsion tizimlarda oqimlar tushunchasi

Oqim (ing. "threads") - bu protsessor tomonidan qayta ishlanadigan buyruqlar ketma-ketligi. Bitta jarayon doirasida bir yoki bir nechta oqim bo'lishi mumkin. Har bir jarayon doirasida bitta oqim qatnashsa bir oqimli yondashuv deyiladi. Masalan, MS-DOS bitta oqimli foydalanuvchi jarayonini qo'llab-quvvatlaydi. Ba'zi UNIX operatsion tizimlarini ko'plab foydalanuvchilarning jarayonlarini qo'llab-quvvatlaydi, lekin bir vaqtning o'zida xar bir jarayon bitta oqimni o'z ichiga oladi oladi. Ko'p oqimlilik (multithreading) - bu OTning bir jarayon doirasida ko'plab oqimlarni qo'llab-quvvatlash qobiliyati deyiladi. Ko'p oqimli tizimlarga Java Runtime Environment, Windows 2000, Linux, Solaris va boshqalarni misol qilish mumkin



6.1.Rasm. Bir oqimli va ko'p oqimli jarayonlar

Parallelizm turlari

Nazariyada ish hajmini parallellashtirishning ikki xil usuli mavjud:

1. **Ma'lumotlar parallelligi** ma'lumotlarni bir nechta yadrolar (oqimlar) o'rtasida taqsimlaydi va har bir kichik ma'lumotlar to'plamida bir xil vazifani bajaradi. Masalan, katta rasmni qismlarga ajratish va turli yadrolarda ishlash.

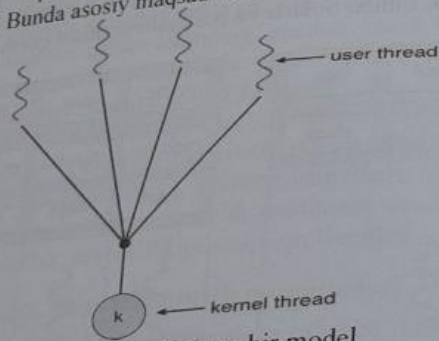
2. **Vazifa paralleligi**, bajarilishi kerak bo'lgan turli xil vazifalarni turli xil yadrolarga ajratadi va ularni bir vaqtning o'zida bajaradi.

6.2. Dasturlarning ko'p oqimli (Multithreading) bajarilishi

Ko'p oqimlilikning turli xil modellari, ya'ni, foydalanuvchi oqimlarini yadro oqimlarida tasvirlash usullari mavjud. Quyidagi multithread modellari nazariy jihatdan amalga oshirish mumkin (va amalda xam mumkin):

“Ko'pga-bir” modeli

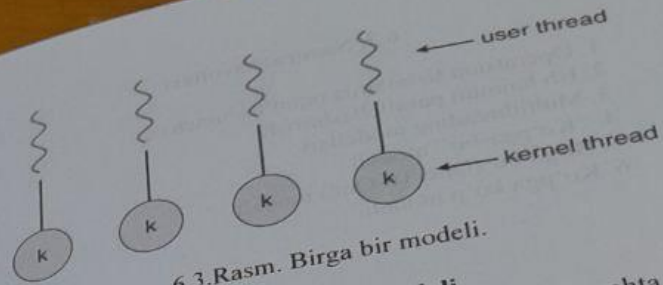
• Ko'pga-bir (many-to-one) modelida ko'plab foydalanuvchi darajasidagi oqimlar bitta yadroli oqimga joylashtirilgan. Tizimli ko'p oqimlilikni qo'llab-quvvatlamaydigan operatsion tizimlarda qo'llaniladi. Bunda asosiy maqsad hotirani tejash.



6.2. Rasm - Ko'pga bir model

Birga bir (One-To-One) modeli

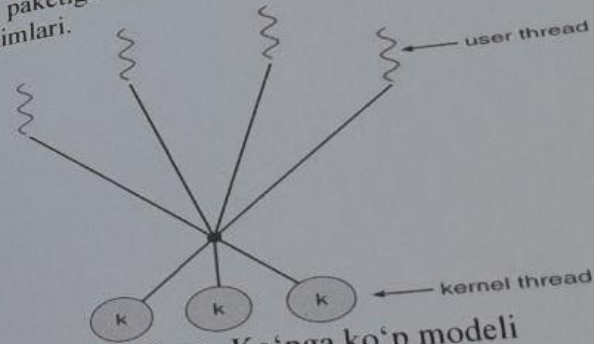
• Birga bir (One-to-one) modeli har bir foydalanuvchi oqimini boshqarish uchun alohida yadro oqimini yaratadi. Windows oilasidagi operatsion tizimlar birga bir modeli asosida ishlaydi.



6.3. Rasm. Birga bir modeli.

Ko'pga ko'p modeli

• Ko'pga ko'p (many-to-many) modeli – bu bir nechta tizim oqimlarida bir nechta foydalanuvchi oqimlarini tasvirlay oladigan model tushuniladi. Bunday model operatsion tizimga ko'p sonli tizim oqimlarini yaratish imkonini beradi. Bunday modelda ishlaydigan operatsion tizimlarga ega Solaris operatsion tizim, shuningdek, ThreadFiber paketiga ega Windows NT/2000/XP/2003/2008/7/10 operatsion tizimlari.



6.4. Rasm. Ko'pga ko'p modeli

Yuqorida ta'qidlab o'tganimizdek, Windows tizimlarida birga bir ko'p oqomlilik modeli amalga oshirilgan:

- oqim identifikatori (thread id);
- registrlar to'plami;
- foydalanuvchi va tizim protseduralari uchun alohida steklar
- oqimning lokal ma'lumotlari uchun hotira xajmi (thread-local storage - TSL)

6.3. Nazorat savollari

1. Operatsion tizimlarda oqim tushunchasi
2. Ish hajmini parallelashtirish
3. Multithreading modellari
4. "Ko'pga-bir" modeli
5. Birga bir (One-To-One) modeli
6. Ko'pga ko'p modeli.

7. OPERATSION TIZIMLARDA TUPIK (DEADLOCK) TUSHUNCHASI

7.1. Tupik muammolari

Kompyuter tizimlarida shunday resurslar borki, bir vaqtning o'zida faqat bitta jarayon orqali ishlatilishi mumkin. Masalan, printerlar, magnit lentaga yozadigan qurilmalarni eslashimiz mumkin. Agar ikkita jarayon bir vaqtning o'zida printerda ma'lumot jo'natsa, u holda chalkashchilik yuzaga keladi. Agar ikkita jarayon fayl tizimi jadvalining bir xil elementidan foydalansa, unda bu tizim albatta buziladi. Shuning uchun barcha operatsion tizimlar muayyan resurslarga vaqtinchalik kirish huquqini berishga qodir. Ko'pgina ilovalarning ishlash printsipida jarayon bir vaqtning o'zida birgina resursga emas, balki bir nechta resurslarga murojaat qila olishi kerak. Faraz qilaylik, ikkita jarayonning har biri skaner qilingan hujjatni Blu-ray diskiga yozmoqchi. A jarayoni skanerdan foydalanish uchun ruxsat oladi. B jarayoni boshqacha dasturlashtirilgan: avval Blu-ray diskidan foydalanish uchun ruxsat oladi. Endi A Blu-ray diskidan foydalanish uchun ruxsat so'raydi, lekin bu qurilma B jarayonidan ozod qilinmaguncha so'rov rad etiladi. Afsuski, bo'shatish o'rniga B jarayon skanerdan foydalanish uchun ruxsat so'raydi. Va shu nuqtada, har ikkala jarayon ham abadiy bloklanadi. Bu holat **berk holat** yoki o'zaro berk xodisa (**deadlock**) deb ataladi. O'zaro berk xodisalar mashinalar o'rtasida ham sodir bo'lishi mumkin. Misol uchun, ofislar ko'plab kompyuterlar ulangan mahalliy tarmoq bilan jihozlangan. Ko'pincha skanerlar, BLU-ray disklari va DVD disklari, printerlar va magnit lenta disklari kabi qurilmalar har qanday mashinada foydalanuvchi uchun mavjud bo'lgan umumiy resurslar sifatida tarmoqqa ulangan bo'ladi. Agar ushbu qurilmalar masofadan boshqariladigan bo'lishi mumkin bo'lsa (masalan, foydalanuvchining uy kompyuteridan), unda yuqorida xodisaga o'xshash o'zaro berk xodisa paydo bo'lishi mumkin. Murakkab bo'lgan sharoitlarda uch, to'rt yoki undan ortiq qurilma va foydalanuvchilar bir-biriga o'zaro berk xodisaga duch kelishlari mumkin. Boshqa holatlarda ham o'zaro berk xodisalar paydo bo'lishi mumkin. Misol uchun, ma'lumotlar bazasini boshqarish tizimlarida, raqobat holatiga tushmaslik uchun, dastur o'zi foydalanadigan bir

nechta yozuvlarni bloklashi kerak bo'lishi mumkin. Agar jarayon A R1 yozuvini bloklay va B jarayoni R2 yozuvini bloklay, keyin har bir jarayon boshqa jarayonning yozuvini bloklayga harakat qilsa, u holda yana o'sha berk holat yuzaga keladi. Shunday qilib, apparat va dasturiy ta'minot resurslari bilan ishlashda o'zaro berk holatlar paydo bo'lishi mumkin.

7.2. Resurslar taqsimlash grafi

O'zaro berk hodisalarning asosiy qismi muayyan jarayonlarga alohida kirish huquqi berilgan resurslar bilan bog'liq. Ular orasida qurilmalar, ma'lumotlar yozuvlari, fayllar va boshqalar mavjud. O'zaro berk hodisalar haqida imkon qadar ko'proq ma'lumotlarga ega bo'lish uchun ob'ektlarni resurslar deb ataymiz. Resurslar apparat qurilmalari (masalan, Blu-ray diskleri) yoki biror ma'lumotlar (masalan, ma'lumotlar bazasi yozuvi) bo'lishi mumkin. Odatda, kompyuterda jarayonga taqdim qilinadigan ko'plab resurslar bo'lishi mumkin. Ba'zi resurslar bir nechta nusxada bo'lishi mumkin, masalan, uchta Blu-ray disk mavjud bo'lishi mumkin. Resursning bir nechta nusxasi mavjud bo'lganda, ulardan biri har qanday resurs so'rovini qondirish uchun ishlatilishi mumkin. Qisqa qilib aytganda, resurs taqdim etilishi, ishlatilishi va bir muncha vaqt o'tgach bo'shatilishi kerak, chunki bir vaqtni ichida faqat bitta jarayon tomonidan foydalanilishi mumkin.

7.3. Bo'shatiladigan va bo'shatilmaydigan resurslar

Resurslar ikki xil bo'ladi: bo'shatiladigan va bo'shatilmaydigan. Egallab turgan jarayondan muammosiz qaytib olinadigan resurslar bo'shatiladigan resurslarga tegishlidir. Bunday resurslarga xotirani misol qilish mumkin. 1 Gb foydalanuvchi xotirasi, bitta printer va xar biri 1 Gb dan ikkita jarayonga ega bo'lgan tizimni ko'rib chiqamiz, Jarayonlarning har biri printerda ma'lumot chop etmoqchi. A jarayoni printerdan foydalanishni so'rab oladi, so'ngra chop etish uchun mo'ljallangan qiymatni hisoblab chiqadi. Ammo hisoblash tugagunga qadar, unga ajratilgan vaqt tugaydi va u diskka yuklanadi. Endi B jarayoni ishga tushadi, printerni ishlatishga harakat qilib muvaffaqiyatsizlikga uchraydi, chunki A jarayoni printerni, B jarayoni

esa xotirani egallab turibdi va ularning hech biri boshqa jarayon tomonidan egallanib turgan resursiz o'z ishini davom ettira olmaydi. Yaxshiyamki, B jarayonidan xotirani olib tashlash imkoniyati mavjud, bu jarayonni diskka tushirish va u yerdan A jarayon yuklab olish mumkin. Endi A o'z ishini davom ettirishi, chop etishi va printerni bo'shatishi mumkin. Natijada hech qanday o'zaro berk hodisa muvaffaqiyatsizliklardan tashqari uning hozirgi egasidan olib bo'lmaydi. Biroq, bo'shatilmaydigan resursni potensial bo'lmaydi. Agar BLU-ray diskda yozishni boshlagan jarayondan yozish qurilmasi olib tashlansa va uni boshqa jarayonga berilsa, Blu-ray disk yaroqsiz holga keladi. Blu-ray disklarini yozish qurilmalarini hoxlagan paytda so'rab olib bo'lmaydi. Resursning bo'shatilishi kontekstga bog'liq. Standart shaxsiy kompyuterda xotira bo'shatilishi uchun ma'lumotlar har doim diskka ko'chirilishi mumkin. Xotirani bo'shatish (swapping) yoki xotirani sahifa usulida tashkil etishni qo'llab-quvvatlamaydigan smartfonda xotira etishmasligi tufayli o'zaro berk hodisadan oson qutilishni iloji yo'q. Odatda, berk hodisa ichida bo'shatilmaydigan resurslar mavjud. Odatda, bo'shatiladigan resurslarni o'z ichiga olgan potensial berk hodisalarni resurslarni bir jarayondan ikkinchisiga qayta taqsimlash yo'li bilan bartaraf etish mumkin. Shuning uchun bizning e'tiborimiz bo'shatiladigan resurslarga qaratiladi.

Ko'pchilik hollarda, resursdan foydalanganda quyidagi voqealar ketma-ketligi sodir bo'ladi:

1. Resurs so'rovi;
2. Resursdan foydalanish;
3. Resursni ozod qilish.

Agar so'rov davomida resurs band bo'lsa, talab qilayotgan jarayon kutish holatiga o'tishga majbur bo'ladi. Ba'zi operatsion tizimlarda, talab qilingan resursni ajratish rad etilsa, jarayon avtomatik ravishda bloklanadi va resurs bo'shaganda - davom etadi. Boshqa tizimlarda talab qilingan resursni ajratishni rad etish xato kodi bilan birga keladi va nima qilish kerakligi haqida qaror qabul qilish, biroz kutish yoki resursni qayta olishga harakat qilish, chaqiruvchi jarayoniga o'tkaziladi. Resursni ajratish talabi rad etilgan jarayon, odatda, qisqa siklga kiritiladi: resurs so'rovi, keyin to'xtatib turish - keyin yana urinit

ko'radi. Bu jarayon berk hodisa qilinmagan bo'lsa-da, lekin barcha ko'rsatkichlar bo'yicha u aslida bloklanadi, chunki u hech qanday foydali ishni amalga oshirmaydi. Muammoni ko'rib chiqish davomida, talab qilingan resursni ajratishdan bosh tortgan holda, jarayon hozirda kutish holatiga tushadi. Resurs so'rovining xususiyatlari ishlatiladigan tizimga sezilarli darajada bog'liq. Ba'zi tizimlarda so'rov uchun request tizimi chaqiruvni taqdim etiladi, Bu jarayonlarga resursni aniq shaklda so'rashga imkon beradi. Boshqa tizimlarda operatsion tizim bilan yagona resurslar ma'lum bir vaqtda faqat bitta jarayon orqali ochilishi mumkin bo'lgan maxsus fayllardir. Ular oddiy Open chaqiruvni yordamida ochiladi. Agar fayl allaqachon ishlatilgan bo'lsa, chaqirilyotgan jarayon fayli joriy egasi tomonidan yopilgunga qadar bloklanadi.

7.4. O'zaro berk hodisalar

O'zaro berk hodisalarga quyidagi rasmiy ta'riflarni berish mumkin. Jarayonlar guruhidagi berk hodisalash, agar ushbu guruhdagi har bir jarayon faqatgina o'sha guruhdan boshqa jarayonga bog'liq bo'lgan hodisalarni kutsa, paydo bo'ladi. Barcha jarayonlar kutish holatida bo'lgani uchun, hech bir jarayon bu guruhga tegishli bo'lgan boshqa jarayonning ishini davom ettirishi mumkin bo'lgan har qanday voqea uchun sabab bo'lmaydi va barcha jarayonlarni kutish cheksiz bo'ladi. Ushbu modelda jarayonlarning faqat bitta oqimi borligi taxmin qilinmoqda va bloklangan jarayonning ishini davom ettira oladigan uzilishlar yo'q. Uzilishlarning yo'qligi holati uchun zarur, boshqa sabablarga ko'ra bloklangan jarayonning o'z ishini davom ettirishiga yo'l qo'ymaslik uchun, masalan, signal orqali, keyin guruhda mavjud bo'lgan boshqa jarayonlarni ozod qiluvchi voqea sabab bo'lishi kerak. Ko'pgina hollarda, har bir jarayonning boshlanishi kutilayotgan voqea, hozirgi vaqtda guruhning boshqa ishtirokchisi bo'lgan har qanday resursni chiqarishdir.

Boshqacha qilib aytganda, bir-biriga bloqlash holatiga tushgan guruhning har bir jarayoni bir xil guruhdan boshqa jarayonga ega bo'lgan resursni kutadi. Hech bir jarayon ishlamaydi, ularning hech biri hech qanday resursni ozod qila olmaydi va ularning hech biri o'z ishini davom ettira olmaydi. Jarayonlar soni va ushlab turilgan va so'ralgan

resurslarning soni va turi muhim emas. Ushbu natija har bir resurslar, jumladan, apparat va dasturiy ta'minot uchun o'zgarmas hisoblanadi. Ushbu turdagi bloklash resurs bloklanishi deb ataladi. Ehtimol, bu eng keng tarqalganidir, ammo yagona turi emas. Dastlab, biz resurs bloklanishini ko'rib chiqamiz.

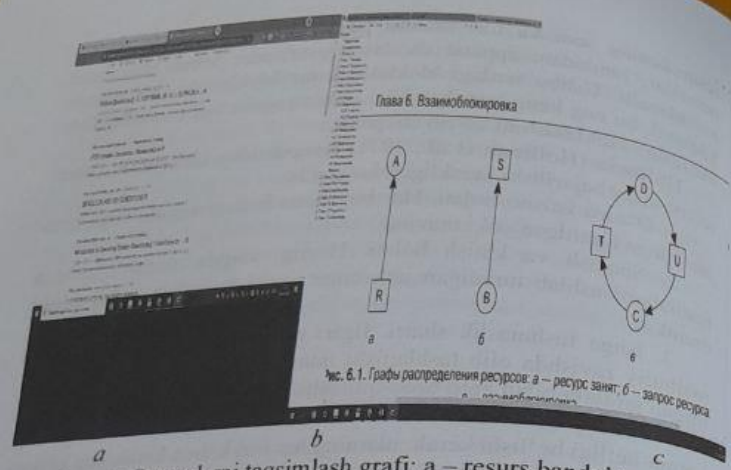
Hoffman (Hoffman et al., 1971) resurslarning bloklanishi uchun to'rtta shart bajarilishi kerakligini ko'rsatdi:

1. O'zaro istisno holati. Har bir resurs hozirgi vaqtda faqat bitta jarayonga ajratilgan yoki mavjud.
2. Saqlash va kutish holati. Hozirgi vaqtda ularga ajratilgan resurslarni ushlab turadigan jarayonlar yangi resurslarni talab qilishi mumkin.
3. Ishga tushmaslik sharti. Ilgari ajratilgan resurslar jarayondan majburiy ravishda olib tashlanishi mumkin emas. Ular ularni ushlab turgan jarayon bilan aniq ozod qilinishi kerak.
4. Davriy kutish holati. Ikki yoki undan ortiq jarayonlarning halqa ketma-ketligi bo'lishi kerak, ularning har biri ketma-ketlikning keyingi a'zosi tomonidan saqlanadigan resursni chiqarishni kutadi.)

Resurs o'zaro berk hodisasi paydo bo'lishi uchun to'rtta shartni barchasi bajarilishi kerak. Agar ulardan biriga amal qilinmasa, manba o'zaro berk hodisasi mumkin emas.

7.5. O'zaro berk hodisalarni modellashtirish

Xolt (Xolt, 1972) yo'naltirilgan graflar yordamida ushbu to'rt shartni qanday modellashtirish mumkinligini isbotlab berdi. Graflarda ikki xil tugun mavjud: doiralalar bilan ko'rsatilgan jarayonlar va kvadratlar bilan ko'rsatilgan resurslar. Resurs tugunidan (kvadratidan) jarayon tuguniga (doirasiga) qarab yo'naltirilgan qirra, ushbu resurs oldindan so'ralganligini, qabul qilinganligini va hozirda ushbu jarayon tomonidan ushlab turilishini anglatadi. R resurs hozirda A jarayoniga ajratilgan (1-rasm, a).



1 rasm. Resurslarni taqsimlash grafi: a – resurs band; b – resurs so'rovi; d – bloklanish

Jarayondan resursga o'tuvchi yo'naltirilgan qirra, bu jarayon hozirda ushbu resurs chiqarilishini kutib, bloklanganligini anglatadi. 1-rasm, b da, B jarayon S resurs chiqarilishini kutmoqda. 1-rasm, c da biz o'zaro berk holatni ko'rib turibmiz: C jarayoni hozirda D jarayoni ushlab turgan T manbasini kutmoqda, D jarayoni T manbasini chiqarmoqchi emas, chunki u ushlab turgan U resursini chiqishini kutmoqda. C jarayoni bilan ushlab turilgan. Ikkala jarayon ham abadiy kutish holatida. Grafikning dumaloq tuzilishi shuni anglatadiki, biz tsikldagi jarayonlar va resurslarni o'z ichiga olgan o'zaro blikorovka bilan ish olib borayapmiz (tizimda har bir turdagi bitta resurs mavjud deb taxmin qilinadi). Ushbu misolda quyidagi tsikl chiqdi: C - T - D - U - C)

7.6. O'zaro berk hodisadan chiqish

Bizning o'zaro berk hodisalarni aniqlash algoritmi muvaffaqiyatli ishlagan va bunday qulfn topdi. Keyingi nima? Bizga chiqib ketishning ba'zi usullari kerak bo'ladi, tizimning ish faoliyatini tiklashga imkon

beradigan. Ushbu bo'limda o'zaro berk hodisadan chiqishning turli usullari ko'rib chiqiladi. Lekin ularning hech biri o'ziga xos e'tirozga ega emas.)

7.7. Tupiklarni qayta ishlash usullari

7.7.1. Resurslarni ustivor o'zlashtirish hisobiga tiklash

Ba'zan siz vaqtincha uning hozirgi egasidan resursni olib tashlash va boshqa jarayonga o'tkazishingiz mumkin. Ko'pgina hollarda, bu, ayniqsa, universal mashinalarda ishlaydigan ommaviy ishlov berish operatsion tizimlarida operatorning aralashuvini talab qilishi mumkin. Misol uchun, lazer printerni egasidan olish uchun operator allaqachon chop etilgan barcha varaqlarni to'plamga qo'yishi mumkin. Keyinchalik, jarayon to'xtatilishi mumkin (ishlamaydigan deb belgilanadi). Shundan so'ng, printer boshqa jarayonga ajratilishi mumkin. Ushbu jarayon o'z ishini tugatgandan so'ng, bosilgan qog'oz joylashtirilishi va asl jarayonning ishlashi qayta tiklanishi mumkin. Jarayondan resursni olib tashlash, uni boshqa jarayonga foydalanishga ruxsat berish va keyin jarayonni ogohlantirmasdan qaytarish qobiliyati asosan ushbu resursning qabul qilish tepsisi ichiga usul bilan tiklash ko'pincha qiyin yoki umuman mumkin emas. To'xtatib turish uchun jarayonni tanlash jarayoni qaysi manbaga ega ekanligi bilan bog'liq bo'lib, uni osongina olish mumkin.

7.7.2. Orqaga qaytarish orqali tiklash

Agar tizim ishlab chiquvchilari va kompyuter operatorlari o'zaro berk hodisa qilish ehtimoli borligini bilsalar, ular nazorat nuqtalarining davriy yaratilishini tashkil qilishlari mumkin. Bu shuni anglatadiki, jarayonning holati faylga yoziladi, va uning qayta ishga tushirilishiga imkon beradi. Nazorat nuqtalari nafaqat xotira tasvirini, balki resurslarning holatini, ya'ni hozirgi paytda qaysi resurslar jarayonga ajratilgani haqida ma'lumot beradi. Keyinchalik samaradorlik uchun yangi nazorat nuqtasi eski faylning ustiga emas, balki yangi faylga

yozilishi kerak. jarayonni bajarish davomida tekshiruv punktlarining butun ketma-ketligi yig'iladi.

Agar o'zaro qulflar aniq aniq, qanday resurslarga ehtiyoj borligini aniqlash oson. O'zaro berk hodisadan chiqish uchun kerakli resursga ega bo'lgan jarayon ushbu resursni olishdan oldingi nuqtaga qaytib, u o'z nazorat punktlaridan birida ishga tushirildi. Ushbu nazorat nuqtasidan keyin amalga oshirilgan barcha ishlar yo'qoladi (masalan, ushbu nazorat nuqtasidan keyin chop etilgan barcha chiqish ma'lumotlari qayta chop etilishi kerak). Aslida, bu jarayon, hozirgi vaqtda o'zaro berk hodisada ishtirok etadigan jarayonlardan biriga ajratilgan resursga ega bo'lmagan avvalgi nuqtaga qaytadi. Qayta ishga tushirilgan jarayon yana resursni olishga harakat qilsa, u mavjud bo'lgunga qadar kutish kerak.

7.7.3 Jarayonlarni yo'q qilish orqali tiklash

O'zaro berk hodisa qilishni to'xtatishning eng qo'pol, ammo eng oson usuli-bir yoki bir nechta jarayonlarni yo'q qilishdir. O'zaro qulflash jarayonida bo'lgan jarayonni yo'q qilish mumkin. Agar omadingiz bo'lsa, boshqa jarayonlar o'z ishlarini davom ettirishi mumkin. Agar bu yordam bermasa, tsikl uzilib qolguncha hamma narsatsh takrorlash mumkin.

Shu bilan bir qatorda, jabrlanuvchi tomonidan tanlangan resurslarni ozod qilish uchun tsiklda bo'lmagan jarayonni tanlash mumkin. Ushbu yondashuv bilan yo'q qilinadigan jarayon juda ehtiyotkorlik bilan tanlanadi, chunki u tsikldagi ba'zi jarayonlar uchun zarur bo'lgan resurslarni saqlab turishi kerak. Misol uchun, bitta jarayon printerni ushlab turishi va plotterni talab qilishi mumkin, ikkinchisi esa plotterni ushlab turishi va printerni so'rashi mumkin. Ularning ikkalasi ham o'zaro qulfl holatida. Uchinchi jarayon boshqa bir printerni va boshqa bir plotterni ushlab turishi va muvaffaqiyatli ishlashi mumkin. Uchinchi jarayonni yo'q qilish bu resurslarni bo'shatilishiga olib keladi va dastlabki ikki jarayonning o'zaro bloklanishini yo'q qiladi.

Iloji bo'lsa, boshidan qayta boshlash mumkin bo'lgan jarayonni o'ldirish yaxshiroqdir. Misol uchun, kompilyatsiya har doim qayta boshlanishi mumkin, chunki u hamma kirish faylini o'qiydi va ob'ekt

faylini yaratadi. Agar kompilyatsiya jarayoni yarim yo'lda vayron bo'lsa, birinchi ishga tushirish ikkinchisiga ta'sir qilmaydi.

Ammo ma'lumotlar bazasini yangilash jarayoni har doim ham xavfsiz tarzda ikkinchi marta ishga tushirilmaydi. Agar jarayon ma'lumotlar bazasi jadvalining har qanday yozuviga birlikni qo'shsa, uning dastlabki ishga tushirilishi, yo'q qilinishi va keyin qayta ishga tushirilishi noto'g'ri natijaga olib keladi, chunki maydonga ikkilik qo'shiladi.

7.8. Tupiklarni oldini olish

Qanday qilib haqiqiy tizimlarda o'zaro qulfdan oldini olish mumkin? Axir biz allaqachon ulardan qochish mumkin emasligiga ishonch hosil qildik, chunki bu hech narsa ma'lum bo'lmagan kelajakdagi so'rovlar haqida ma'lumot talab qiladi. Bu savolga javob berish uchun to'rtinchi shartga qaytamiz. Hoffman va uning hamkasblari tomonidan shakllantirilgan (Kofman at Al., 1971) va bizni muammoni hal qilish uchun kalit bera olishini kuzatamiz? Agar biz ushbu shartlardan kamida bittasi hech qachon bajarilmasligini kafolatlay olsak, o'zaro berk hodisalarni tizimli ravishda amalga oshirish mumkin bo'lmaydi (Havender, 1968).

Qotib qolish

Oddiy va faol berk hodisalash bilan chambarchas bog'liq bo'lgan muammo qotib qolishdir. Dinamik tizimda resurslar sorovi doimiy ravishda amalga oshiriladi. Qaysi resursni kim va qachon qabul qilishi haqida qaror qabul qilish uchun sizga ma'lum bir siyosat kerak. Ushbu siyosat, hatto oqilona bo'lsa ham, ular bir-biriga qulflash holatida bo'lmasa ham, ba'zi jarayonlar hech qachon xizmat qilmasligiga olib kelishi mumkin.

Misol sifatida, printerni taqsimlashni ko'rib chiqing. Tasavvur qilamiz, tizim printerni taqsimlashni o'zaro qulflashga olib kelmasligini ta'minlaydigan ma'lum bir algoritmdan foydalanadi. Keling, bir necha jarayonlar printerni o'z ixtiyorida olishni istashgan deb taxmin qilaylik.

Mumkin bo'lgan algoritmlardan biri printerni chop etish uchun eng kichik faylga ega bo'lgan jarayonga o'tkazishni o'z ichiga oladi (bunday ma'lumot mavjud deb hisoblanymiz). Ushbu yondashuv maksimal darajada baxtli mijozlar sonini oshiradi va juda adolatli ko'rinadi. Keling, bir jarayonning chop etish uchun katta faylga ega

bo'lgan ishlaydigan tizimda nima sodir bo'lishini ko'rib chiqaylik. Printer yana bir marta ozod bo'lgach, tizim tekshiriladi va eng qisqa fayl bilan jarayonni tanlaydi. Qisqa fayllardagi jarayonlar oqini tugamasa, katta fayl jarayoni printerni hech qachon qabul qilmaydi, u faqat qattiq qotib qoladi (hatto berk hodisa qilinmasa ham, abadiy qoldiriladi).

Resurslarni taqsimlash siyosatini qo'llash orqali qotib qolishni to'xtatish mumkin "birinchi bo'lib kelganga-birinchi qolishni ko'rsatiladi asosida". Ushbu yondashuv bilan, eng uzoq davom etayotgan jarayonga xizmat ko'rsatiladi. Oxir-oqibat, har qanday berilgan jarayon oxir-oqibat navbatning eng kattasiga aylanadi va kerakli resursni oladi.

Shuni ta'kidlash kerakki, ba'zilar qotib qolishni va o'zaro berk holatni farqlamaydi, chunki har ikkala holatda ham oldinga harakat yo'q. Boshqalar esa asosiy farqni his qilishadi, chunki jarayon osonlik bilan n marta biror narsa qilishga harakat qilish uchun dasturlashtirilishi mumkin va agar barcha urinishlar muvaffaqiyatsizlikka uchrasa, boshqa narsalarni qilishga harakat qilinadi. Bloklangan jarayon esa bunday imkoniyatga ega emas.

7.9. Nazorat savollari

1. Operatsion tizimlarda tupik (deadlock) tushunchasi
2. Boshi berk jarayonlar
3. Resurslar
4. Bo'shatiladigan va bo'shatilmaydigan resurslar
5. O'zaro berk hodisalar
6. O'zaro berk hodisalarni modellashtirish
7. O'zaro berk hodisadan chiqish
8. Resursni ustuvor o'zlashtirish hisobiga tiklash
9. Orqaga qaytarish orqali tiklash
10. Jarayonlarni yo'q qilish orqali tiklash
11. O'zaro berk hodisaning oldini olish
12. Qotib qolish

8. OPERATSION TIZIMDA XOTIRANI BOSHQARISH

Kompyuter tizimi bosh asosiy masalasi dasturni boshqarishdir. Dasturlar va ularning murojaat qiladigan ma'lumotlari, bajarilish jarayonida operativ xotirada (xech bo'lmasa qisman) joylashgan bo'lishi shart. Operatsion tizimga xotirani foydalanuvchi jarayonlari va OT komponentalari orasida taqsimlashga to'g'ri keladi. Operatsion tizimning bu faoliyati xotirani boshqarish deyiladi. Shunday qilib, xotira (storage memory) sinchiklab boshqarishni talab etadigan resursdir. Yaqin kunlarga xotira eng qimmat resurs xisoblangan. Operatsion tizimning, xotirani boshqaradigan qismi, xotira menejeri deyiladi.

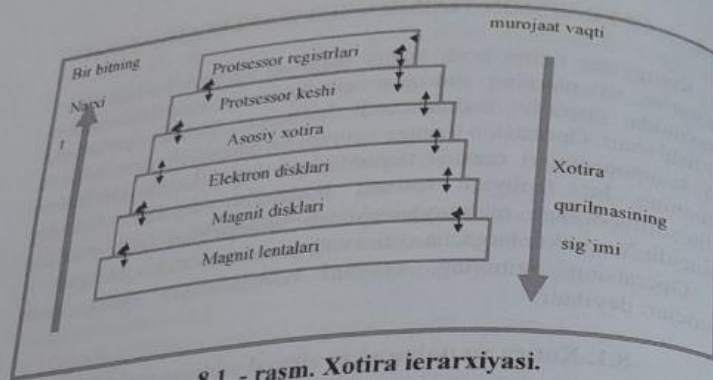
8.1. Xotira va uning aksi, virtual address makoni

Kompyuterning hotira qurilmasi ikki xil turga: asosiy (bosh xotira, tezkor xotira, fizik xotira) va ikkilamchi (ichki xotira) xotiraga bo'linadi.

Asosiy xotira bir baytli tartiblangan yacheyka massiviga ega bo'lib, har bir yacheyka o'zining adresiga (nomeriga) ega. Protessor buyruqlarni asosiy xotiradan oladi, qayta ishlaydi va bajaradi. Buyruqlarni bajarishda asosiy xotiraning bir nechta yacheykalariga murojaat qilishga to'g'ri keladi. Odatda asosiy xotira yarimo'tkazgichli texnologiya asosida tayyorlanadi shuning uchun xotiradagi ma'lumotlar elektr manbasidan uzilgandan so'ng o'chib ketadi.

Ikkilamchi xotira (bu asosan disklardir) bu chiziqli birlik adresga ega bo'lgan joy va ularni ketma-ket joylashgan baytlar tashkil qiladi. Ikkilamchi xotiraning tezkor xotiradan farqi shundaki, u alohida energiyaga, katta hajmga, va samarali foydalanish imkoniyatiga ega.

8.1 rasmdagi ko'rsatilgan sxemaga yana bir nechta oraliq satxlarni qo'shish mumkin. Xar xil ko'rinishdagi xotiralar ierarxiyaga, murojaat vaqti kamayib borishi, narxini oshishi va sig'imi oshishi tarzida birlashishi mumkin.



8.1. - rasm. Xotira ierarxiyasi.

Ko'p bosqichli sxemalar quyidagicha ishlatiladi. Ma'lumotlar odatda xotiraning yuqori satxlaridan qidiriladi, agar u yerdan topilmasa, ma'lumotlar katta nomerli satxlarda xam saqlanadi. Shuning uchun, u keyingi satxdan qidira boshlaydi. Agar kerakli ma'lumotni topsa, uni yuqoriroq satxga o'tkazadi.

Maxallilik (lokallilik)

Ma'lum bo'lishicha, bu usulda boshqarishni tashkil etish xotira satxlariga kirishni va aloqa chastotasini kamaytiradi.

Bu yerda muxim rolni, chegaralangan vaqt davomida, xotira adreslarining kichik bo'lagi bilan ishlash xossasi o'ynaydi. Bu empirik jixatdan kuzatiladigan xossa lokallilik prinsipi yoki murojaatlarni lokallashtirish deyiladi.

Protessor KEShi, qurilmalarning bir qismi xisoblanadi, shuning uchun OTning xotira menejeri, asosan ma'lumotlarni kompyuterning asosiy va ichki xotira qismiga taqsimlash bilan shug'ullanadi. Ba'zi sxemalarda tezkor va ichki xotira o'rtasidagi oqimni dasturchi boshqaradi. Ammo bu bog'lanish dasturchi vaqtini yo'qotadi, shu sababli bu ishni OT ga yuklashga harakat qilinadi.

Fizik xotirada ma'lumotlarni real joylashishini ko'rsatuvchi asosiy xotiradagi adreslar- fizik adreslar deb ataladi. Dastur ishlaydigan fizik adreslar to'plami, fizik adreslar maydoni deb ataladi.

Mantiqiy (logik) xotira

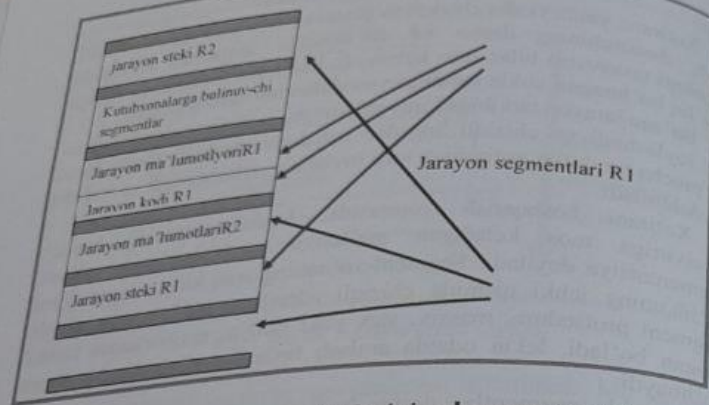
Xotirani, yacheykalar chiziqli to'plami ko'rinishida apparat tashkil etish, dasturchining dastur va ma'lumotlar saqlanishi ko'rinishi xaqidagi tassavvuri bilan mos kelmaydi. Ko'pgina dasturlar bir-biriga bog'liq bo'lmagan xolda yaratilgan modullardan tashkil topgan.

Ba'zan jarayon tarkibiga kiruvchi hamma modullar xotirada ketma-ket joylashadi va chiziqli adreslar maydonini tashkil qiladi. Biroq ko'pincha modullar xotiraning turli joylarida joylashtiriladi va turlicha foydalaniladi.

Xotirani boshqarish sxemasida, foydalanuvchining bunday tassavuriga mos keladigan ma'lumot va dasturlarni saqlash, segmentatsiya deyiladi. Segment-xotiraning aniq ko'rsatilagan qismi bo'lib, uning ichki qismida chiziqli adreslarni qo'llab quvvatlaydi. Segment protsedura, massiv, stek yoki skalyar miqdorlardan tashkil topgan bo'ladi, lekin odatda aralash tipdagi ma'lumotlardan iborat bo'lmaydi.

Boshida segmentlar dastur kodi fragmentlarini (matn redaktori, trigonometrik kutubxona va x.k.) jarayonlar bilan umumlashtirish zaruriyatidan kelib chiqqan bo'lishi kerak, chunki ularsiz xar bir jarayon o'zining adres makonida ma'lumotlarning yana bir nusxasini saqlashiga to'g'ri kelar edi. Xotiraning, tizim bir nechta jarayonning ma'lumotlarini aks ettiradigan aloxida qismlari bo'lib ular segmentlar deb nom oldi.

Xotira shunday qilib, chiziqli ko'rinishdan ikki o'lchamli ko'rinishga keldi. Adres ikki komponentdan iborat bo'lib, ular: segment nomeri va segment ichidagi joylashgan o'rnidir. Keyinchalik, jarayonning turli komponentlarini (dastur kodi, ma'lumotlar, stek va x.k.) turli segmentlarda joylashtirish qulay bo'lib qoldi. Yana shu narsa aniq bo'lib qoldiki, aniq segment ishini, unga segmentda saqlanadigan ma'lumotlar ustida bajarilishi ruxsat berilgan operatsiyalar, masalan, murojaat xuquqi va operatsiyalar tipi kabi atributlar qiymatini berib nazorat qilish mumkin bo'lib qoldi.



8.2. - rasm. Jarayon segmentlarining kompyuter xotirasida joylashishi

Ba'zi jarayonni adres makonini tasvirlaydigan segmentlar 8.2 - rasmda ko'rsatilgan.

Aksariyat zamonaviy operatsion tizimlar xotirani segment boshqaruv xususiyatiga ega. OTlarning ba'zi arxitekturalarida (masalan Intel) segmentlash qurilmalar tomonidan qo'llanadi.

Jarayon murojaat qiladigan adreslar, operativ xotirada mavjud bo'lgan real adreslardan shu taxlitda farq qiladi har bir aniq holatda dastur foydalanadigan adres, har xil usullar yordamida tasvirlanishi mumkin. Masalan, adres berilgan matnda odatda simvolli bo'ladi. Kompilyator bu simvolli adres va o'zgaradigan adresslarni boholaydi (masalan, n bayt modul boshidan). Dastur generatsiyalagan bunday adres odatda mantiqiy adres (virtual xotirali tizimlarda u ko'pincha virtual xotira) deb nomlanadi. Barcha mantiqiy adreslar to'plami mantiqiy (virtual) adreslar maydoni deb ataladi.

Adresslar bog'lanishi

Demak, mantiqiy va fizik adreslar maydonlari, tashkil etilishi va o'lchami bo'yicha bir biriga mos emas. Mantiqiy adreslar maydonning maksimal o'lchami odatda protsessorning razryadi bilan aniqlanadi

(masalan 232), va zamonaviy tizimlarda fizik adreslar maydonining hajmidan ko'zga ko'rinarli darajada yuqori bo'ladi. Shunday ekan, protsessor va operatsion tizim asosiy xotirada joylashgan dasturni kerak. Bunday ko'rinishda adreslarni tasvirlash adreslarni translatsiyasi yoki adreslarni bog'lash deb nomlanadi.

Mantiqiy adressning fizik adres bilan bog'lanishi dastur operatorining bajarilishigacha yoki bajarilish vaqtida amalga oshirilishi shart. Bunday holda, ko'rsatmalarni va ma'lumotlarni xotiraga bog'lash quyidagi qadamlar bo'yicha amalga oshiriladi.

- Kompilyatsiya bosqichi
- Yuklash bosqichi
- Bajarilish bosqichi.

8.2. Xotirani boshqarishning umumiy tamoyillari

Xotiradan samarali foydalanishni ta'minlash uchun operatsion tizim quyidagi funksiyalarni bajarishi lozim:
Fizik xotirani aniq bir sohasida jarayon adreslar to'plamini aks ettirish;

- qarama-qarshi jarayonlar o'rtasida xotirani taqsimlash;
- Jarayonlar adreslar maydoniga ruxsatni boshqarish;
- Operativ xotirada joy qolmaganda, Tashqi xotiraga jarayonlarni (qisman yoki to'liq) yuklash;
- Bo'sh va band xotirani hisobga olish.

8.2.1. Xotira boshqaruvining eng oddiy sxemalari

Dastlabki operatsion tizimlarda xotirani boshqarishning eng oddiy metodlari qo'llanilgan. Boshida foydalanuvchining har bir jarayoni asosiy xotiraga ko'chirilishi kerak bo'lgan, xotira uzluksiz maydonini band qilgan, tizim esa qo'shimcha foydalanuvchi jarayonlarga bir vaqtning o'zida asosiy xotirada joylashib turganicha xizmat ko'rsatadi. Keyin "oddiy svoping" (avvalgi tizimlardagidek xar bir jarayonni asosiy xotiraga joylashtiradi, ammo ba'zi ko'rsatkichlarga asosan ba'zi jarayonlar obzorini asosiy xotiradan tashqi xotiraga to'liq chiqaradi va uni boshqa jarayon obrazi bilan almashtiradi) paydo bo'l

Bu turdagi sxema nafaqat tarixiy qiymatga ega, hozirgi vaqtda ular o'quv-mashq va ilmiy – tajriba modelli operatsion tizimlarida, shuningdek operatsion tizim «tikilgan» xolda o'rnatilgan kompyuterlarda(embedded) qo'llaniladi.

8.2.2. Qat'iy belgilangan(fiksirlangan) bo'limli sxemalar

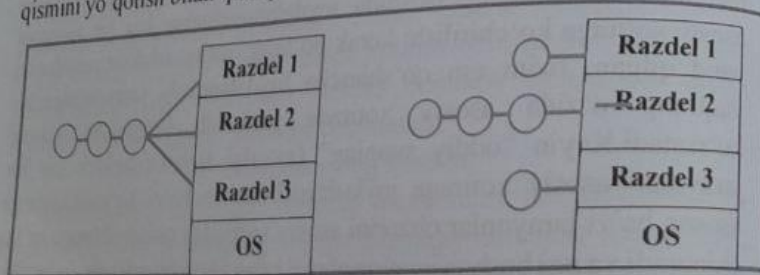
Tezkor xotirani boshqarishning eng oddiy yo'li uni oldindan(generatsiya bosqichida yoki tizim yuklanishi vaqtida) bir qancha qat'iy belgilangan o'lchamdagi bo'limlarga bo'lishdan iboratdir.

Kelib tushayotgan jarayonlar u yoki bu bo'limga joylashtiriladi. Shu sababli fizik adreslar maydonining shartli bo'linishi yuzaga keladi. Jarayonning mantiqiy va fizik adreslari bog'lanishi uni aniq bir bo'limga yuklash vaqtida yoki ba'zan kompilatsiya vaqtida yuzaga keladi.

Har bir bo'lim o'zining jarayonlar navbatiga ega, yoki hamma bo'limlar uchun jarayonlar global navbati mavjud bo'lishi mumkin. Bu sxema IBM OS/360 (MFT), DEC RSX-11 va shunga yaqin boshqa sistemalarda qo'llanilgan. Xotirani boshqarish tizimi jarayonni hajmini baholaydi, unga mos keluvchi bo'limni tanlaydi, jarayonni bu bo'limga yuklaydi va adreslarni sozlaydi.

4 rasmda fiksirlangan bo'limli sxemalar ko'rsatilgan:(a) navbati umumiy bo'lgan jarayonlar,(b)-aloxida navbatli jarayonlar

Bu sxemaning kamchiligi ko'rinib turibdiki, bir vaqtda bajariladigan jarayonlar soni bo'limlar soni bilan cheklangan. Boshqa muhim kamchiligi shundan iboratki, taklif qilinayotgan sxema, ichki fragmentlashdan, ya'ni jarayonga ajratilgan, ammo ishlatilmagan xotira qismini yo'qotish bilan qattiq zararlanadi.



8.3.-rasm Overleyli(qoplangan) tuzilish

Fragmentatsiya, jarayon o'ziga ajratilgan bo'limni to'liq band qilmasligi yoki ba'zi bo'limlar, bajariladigan foydalanuvchi dasturlari uchun kichik bo'lganligidan kelib chiqadi.

Jarayon mantiqiy adreslar maydoni hajmi, unga ajratilgan bo'lim hajmidan katta (yoki eng katta hajmdan ham katta) bo'lgan holatlarda, texnikadan foydalaniladi.

Asosiy g'oya – faqat ayni vaqtda kerak bo'lgan dastur ko'rsatmalarini xotirada saqlab turishdir.

Overlay tuzilish kodining diskda aniq xotira ishlatadi. Overlay kerakli vaqtda uni drayver orqali o'qib ishlatiladigan hamma fayllar kerakli vaqtda uni drayver orqali o'qib ko'rinishdagi fayl yordamida to'ldiriladi.

Shuni nazarda tutish kerakki, overlayli strukturani tashkil etish ko'p jihatdan lokallilik xossasiga bog'liqdir, bu esa o'z navbatida xotirada ayni vaqtda faqat kerak ma'lumotlarni saqlash imkonini beradi.

8.3. Xotirani statik va dinamik taqsimlash, segmentli, sahifali, segment – sahifali tashkil etish

Multidasturli operatsion tizim tomonidan katta e'tibor bilan boshqariladigan resurs bu- xotira. Operatsion tizim bilan egallanmagan operativ xotira taqsimlanishi shart va zarur. Operatsion tizim eng kichik adresslarda joylashadi, lekin eng katta adresslarni egallashi mumkin. Operatsion tizim xotirani asosiy funksiyalari, bo'sh yoki band xotirani aniqlash, jarayonlarga xotirani ajratish va jarayon tamom bo'lgandan keyin xotirani bo'shatish, jarayonlarni operativ xotiradan diskka o'tkazish, agarda asosiy xotirani hajmi yetarli bo'lmasa va xotiradan operatsion xotiraga qayta yuklash, qachonki joy bo'sh va dasturning adresslari aniq tizimiy xotira atrofiga sozlash. Doimiy xotira qurilmasi – ROM. Ushbu xotira faqat o'qib olish uchun mo'ljallangan, bu maxsus qurilma ishlab chiqarish davomida ushbu qurilmaga ma'lumot yoziladi.

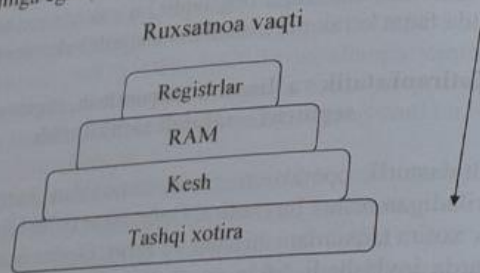
Operativ xotira – RAM tezkor xotira qurilmasi odatda protsess bilan o'zaro aloqada uning vazifasini bajarib tez dastur

ma'lumotlarni yozish, o'qish va saqlash. Uni saqlash vaqti faqat qurilmaga ulangan bo'lishi kerak.

Hisoblash mashinasining xotirasi bu xotira qurilmalarining shajarasi. Ushbu xotiralar o'rtacha ruxsatnoma vaqti bilan va ma'lumot saqlash vaqti bilan 1 bit ma'lumotga nisbatan farq qiladi. Odatda foydalanuvchi arzon va tez xotiraga ega bo'lishni xohlaydi.

Kesh xotira bizlarga ushbu muammoni yechishga yordam beradi.

Kesh xotira bu 2 turdagi xotira qurilmaning o'zaro ishlashini tashkillashtirish usuli, agarda ushbu xotiralar ruxsatnoma vaqti va ma'lumotlar saqlash vaqti bilan bir - biridan farq qilsa kesh o'rtacha vaqtini kamaytirib beradi. Bunga sabab tezkor xotira qurilmaga ma'lumotlarni dinamik xolatda nusxalashdir. Odatda tezligi past xotirada olingan ma'lumotlarni kesh xotirada nafaqat 2 tur xotira qurilmani ishini tashkillashtirish usulini balki tezkor xotira qurilma deb ataydilar. Kesh xotira qurilma odatda qimmatbaho qurilma va kichkina ma'lumot hajmga ega. (4 Mb)



Kesh xotirani qo'llash mexanizmi oddiy foydalanuvchi uchun shaffof, demak kesh xotira ma'lumotlarni ishlatish chastota haqida hech qanday axborotni hech qaysi qurilmaga o'tkazmaydi va hech qaysi turdagi xotira qurilmaga ushbu ma'lumotni o'tkazmasligi shart.

Kesh xotirali tizimlarda operativ xotiraga har bir so'roq keltirilgan algoritim bilan bajariladi. Kesh xotirasi jamlanmasi kerakli ma'lumotlar kesh xotirada bor yo'qligini tekshiradi, shu sababdan kerakli ma'lumotlar qidiruvi operativ xotirani adresi ustun yacheykalaridan arab chiqiladi. Agarda kerakli ma'lumot kesh xotirada bo'lsa ular qib olinadi va natijasi protsessorga o'tkaziladi. Agar kerakli ma'lumotlar bo'lmasa ushbu ma'lumotlar adresi bilan birga operativ

xotiradan kesh xotiraga ko'chirilib olinadi va so'roq bajarish natijalari protsessorga o'tkaziladi. Agarda kesh xotirada bo'sh joy yo'q bo'lsa shu xotiradan oxirgi vaqt mobaynida eng kam murojaat qilingan ma'lumotlar tanlab olinadi va xotiradan chiqariladi, lekin agarda ushbu chiqarilgan ma'lumotlar kesh xotirada joylashgan paytida o'zgartirilgan bo'lsa ular operativ xotiraga ko'chirib olinadi. Agarda ushbu ma'lumotlar o'zgartirilmagan bo'lsa ushbu kesh xotiraning joyi bo'sh deb e'lon qilinadi.

Amalda kesh xotirada bitta ma'lumot element o'qib chiqiladi.

$$t = t_1 + (1 - p) Q t_2(p)$$

t_1 - bu umumiy o'rtacha ma'lumotlarga ruxsat olish tezligi (OZU)

t_2 - bu kesh xotiraga ruxsatnoma vaqti

$$t_2 < t_1$$

t - kesh xotirali tizimning ruxsatnoma olish o'rtacha vaqti

p - ushbu ma'lumot keshda joylashganlik ehtimolligi.

To'liq ehtimollik yuqoridagi formuladan topiladi

Ushbu formula bo'yicha o'rtacha ma'lumotga ruxsatnomani vaqti agar kesh xotirali bo'lsa aniq va chiziqli keshda ma'lumotni tushish va asosiy xotira qurilmaga o'rtacha ruxsatnoma vaqtiga bog'liq. Real tizimlarda kerakli ma'lumotdan keshda joylashgan ehtimollik, $p \approx 0,9$. Ushbu ehtimollikning real tizimlarda shunchalik katta qiymati sababi u har bir ma'lumotda ob'yektiv xususiyati: ushbu xususiyatning nomlari fazo va vaqtli lokalligi.

- Fazo lokallik xususiyati agarda ma'lum bir adressga murojaat qilingan bo'lsa demak yaqin vaqt orasida qo'shni adresslarga murojaat qilish ehtimolligi oshadi.

- Vaqt lokalligi. Agarda ma'lum bir adressga murojaat qilingan bo'lsa demak ushbu adressga yaqin vaqt orada qayta murojaat qilish ehtimolligi ortadi

Yuqorida keltirilgan xulosalar boshqa xotira qurilmalar juftligiga ham mosdir.

Elektron disk bilan ma'lumot almashinish tezligini oddiy qattiq disklarga nisbatan katta. Shu sabab ishlash koeffitsienti oshishi mumkin.

To'g'ri ruxsatnoma DMA(Direct Memory Access)

Har bir tizimli qurilma agarda xotiraga to'g'ri ruxsatnomaga ega bo'lsa kerakli vaqtda adress va ma'lumotdan blokning hajmi bilan

so'roq yuborishi mumkin. Hamda boshqaruv signallari ham yuboriladi. DMA lar bo'yicha xotiraga bir nechta qurilma ruxsatnomalarga ega bo'lishi mumkin. (masalan: protsessor, video karta, qattiq disk)

8.3.1. Dinamik taqsimlanish. Almashtirish (svoping)

Paketli tizimlar bilan ishlashda fiksirlangan bo'limlar bilan ishlab, boshqa xech qanday murakkab narsalardan foydalanmaslik xam mumkin. Vaqtni taqsimlash tizimlari bilan ishlash vaqtida, xotira xamma foydalanuvchilar jarayonlarini o'zida ushlab tura olmaydigan holat ro'y berishi mumkin. Bu holda svopingdan foydalanishga to'g'ri keladi. Svoping-bu jarayonlarni asosiy xotiradan diskka va orqaga to'liq o'tkazishdir. Jarayonlarni diskka qisman yuklash saxifali tashkil etilgan tizimda amalga oshiriladi.

Yuklangan jarayonlar xuddi o'sha adresslar maydoniga yoki boshqa joyga qaytarilishi mumkin. Bu cheklash bog'lanish metodi xususiyatidan kelib chiqadi. Bog'lanish sxemasi uchun, bajarish bosqichida jarayonlarni xotiraning boshqa joyiga ko'chirish mumkin. Svoping xotirani boshqarishga bevosita aloqasi yo'q, u ko'proq jarayonlarni rejalashtirish bilan bog'liqdir.

8.3.2. O'zgaruvchan bo'limli sxemalar

Qoida bo'yicha svoping tizimi fiksirlangan bo'limlarga asoslanishi mumkin. Ammo dinamik taqsimlash yoki o'zgaruvchi bo'limli sxemalar samarali xisoblanadi. Chunki ular xamma jarayonlar to'liq ravishda xotirada joylashganda, yani svoping bo'lmagan xollarda qo'llaniladi.

Bu holda, boshida xotira butunlay bo'sh va oldindan bo'limlarga bo'lingan emas. Yangidan kelayotgan masalaga qat'iy ravishda kerakli xotiraning o'zi ajratiladi. (undan ko'p emas)

Jarayon chiqarilgan dan so'ng, xotira vaqtincha bo'shatiladi. Bir qancha vaqt o'tgandan so'ng xotira turli o'lchamdagi o'zgaruvchili sonli bo'limlardan iborat bo'lib qoladi. Yonma-yon bo'lgan bo'sh joylar birlashtirilishi mumkin.

8.3.3. Saxifali xotira

Yuqorida tavsiflangan sxemalarda xotiradan samarali foydalanilmaydi, shuning uchun xam xotirani taqsimlashning zamonaviy sxemalarida jarayonni operativ xotirada uzluksiz blok sifatida joylashtirish ko'zda tutilmagan.

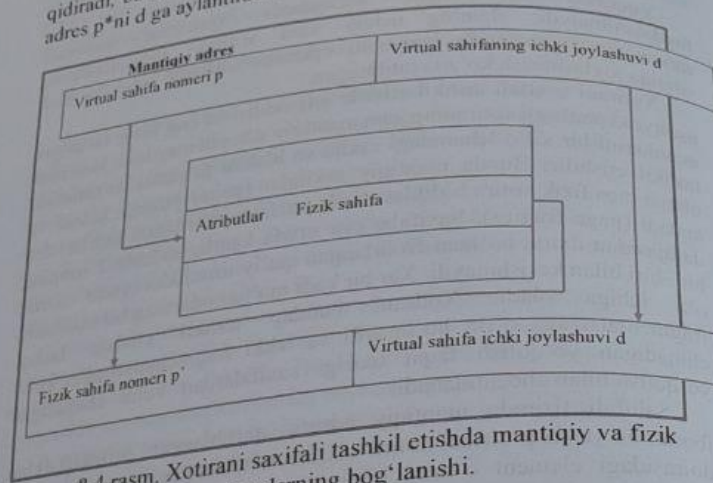
Xotirani saxifali tashkil etishda eng oddiy va eng keng tarqalgan usul (yoki paging), xotiraning xam mantiqiy adressli maydoni, xam fizik maydonini bir xil o'lchamdagi tashkil etishdir. Bunda mantiqiy adressli maydoni, xam fizik tashkil etishdir. Bunda mantiqiy adressli maydoni, xam fizik ularga mos fizik xotira birliklari-fizik saxifalar (page) yuzaga keladi va ataladi (page frames). Saxifalar (va saxifa kadrlari) odatda 2 sonning darajasidan iborat bo'lgan fiksirlangan qat'iy uzunlikka egadir va ular bir-biri bilan kesishmaydi. Xar bir kadr ma'lumotlarning bir saxifasini o'z ichiga oladi. Xotirani bunday tashkil etishda tashqi fragmentatsiyaga o'rin bo'lmaydi va ichki fragmentatsiyadan kelib chiqadigan yo'qotish bilan chegaralanadi.

Sahifali tizimda mantiqiy adress- tartiblangan juftlik (p,d) dan iborat, bu yerda p virtual xotira saxifasi tartib raqami, d esa bu saxifa doirasidagi element o'rnini bildiradi. Adress makonini saxifalarga bo'lish xisoblash tizimi tomonidan dasturchi aralashmagan xolda amalga oshiriladi. Shuning uchun xam adress operatsion tizim nuqtai nazaridagina ikki o'lchamlidir, dasturchi nuqtai-nazaridan esa jarayon adress makoni chiziqli xisoblanadi.

Yuqorida keltirilgan sxema jarayonlarni to'liq joylashtirish uchun kadrlarning uzluksiz soxasi yetarli bo'lmagan xollarda xam jarayonni yuklash imkonini beradi. Ammo, bu sxemada adresni translyatsiyalash uchun bitta asos registri yetarli emas. Mantiqiy adresslarni fizik adresslarda aks ettirish, mantiqiy saxifalarni fizik saxifalarda aks ettirishga keltiriladi va operativ xotirada saqlanadigan saxifalar jadvalidan iborat bo'ladi. Ba'zida, saxifalar jadvali -jadval ko'rinishidagi chiziqli -bo'lakli funktsiya xam deyiladi.

Mantiqiy manzilning interpretatsiyasi 7.4 rasmda ko'rsatilgan. Bunda bajariladigan jarayon vq(p,d) mantiqiy adresga murojaat qiladi, va aks ettirish mexanizmi saxifa tartib raqami r ni saxifalar jadvalidan

qidiradi, bu sahifa r*saxifa kadrda joylashganligini aniqlaydi va real adres p*ni d ga aylantiradi.



8.4 rasm. Xotirani saxifali tashkil etishda mantiqiy va fizik adresslarning bog'lanishi.

Sahifalar jadvali (page table) protsessorning maxsus registrida manzillashtiriladi va kadrlar nomerini mantiqiy adres bo'yicha aniqlashga yordam beradi. Bu asosiy masaladan tashqari saxifalar jadvali qatorida yozilgan atributlar yordamida aniq saxifaga murojaat nazorati va uni ximoyasini tashkil etish mumkin.

8.3.4. Segmentli va segment – sahifali xotira

Xotiralarni boshqarishni yana ikkita sxemasi mavjud: segmentli va segment – sahifali. Segmentlar sahifalardan farqli ravishda o'zgaruvchi o'lchamga ega bo'lishadi. Xotirani segmentli tashkil etishda virtual adres, dasturchi uchun xam, operatsion tizim uchun xam ikki o'lchamli bo'ladi va ikki maydondan: segment tartib raqami va segment ichidagi joy raqamidan iborat bo'ladi.

Aytib o'tish lozimki, tasvirlash qulay bo'lishi uchun OT yordamida chiziqli adres ikki o'lchamlilikka keltirilgan saxifali tashkil etishdan farqli ravishda, bu yerda adres ikki o'lchamliliigi foydalanuvchini jarayonni baytlarning chiziqli massivini ko'rinishida emas, balki o'zgaruvchi uzunlikdagi segmentlar to'plami ko'rishida tassavur qilish natijasidir. (ma'lumot, kod, stek, ...)

Quyida bosqichdagi dasturlash tillarida dastur tuzuvchi dasturchilar segmentli struktura haqida bilishlari lozim. Bunda segment registrini o'zgartirishga olib keladi. Mantiqiy manzillar maydoni – segmentlar yig'indisi. Har bir segment bitta manzil beradigan sahifalarning nomeriga ega. Dasturchi bitta manzil beradigan sahifalarning nomeriga bo'linadigan sahifali sxemalardan farqli ravishda segmentli strukturada manzil ikkita o'lchamdan iborat bo'ladi: segment nomi va joy o'zgarishi. Har bir segment – 0 dan boshlanadigan manzillarning to'g'ri chiziqli ketma-ketligidir. Segmentning maksimal razmeri protsessorning razryadi bilan aniqlanadi. (bunda 32 razryadda manzillashda 232 bayt yoki 4 Gb). Segment manzili dinamik ravishda o'zgarishi mumkin (masalan stek segmenti). Segmentlar jadvalining elementida segment boshlanishi fizik manzilidan tashqari odatda segment uzunligi xam beriladi.

8.4. Nazorat savollari

1. Kompyuter fizik xotirasi qanday strukturaga ega.
2. Mantiqiy xotira va segmentlash tushunchasi.
3. Xotirani boshqarish tushunchasi.
4. Qat'iy belgilangan bo'limli sxemani ayting.
5. Overlayli struktura va dinamik taqsimlash.
6. Xotirani sahifali, segmentli va segment -sahifali tashkil etish.
7. Virtual xotira tushunchasi.

9. VIRTUAL XOTIRA

Virtual xotira kompyuter uchun qo'shimcha xotirani ta'minlash uchun kompyuterning qattiq diskidan foydalanish usuli hisoblanadi. Xotira segmentlari sahifalar sifatida tanilgan qattiq diskda saqlanadi. Virtual xotirada saqlanadigan xotira segmenti so'ralganda, u haqiqiy xotira manziliga yuklanadi.

Virtual xotiraning asosiy afzalliklari dasturlarni umumiy xotira maydonini boshqarish imkoniyatidan ozod qilish, kutubxonalar tomonidan foydalaniladigan xotirani jarayonlar o'rtasida bo'lishish qobiliyati, xotirani ajratish sababli xavfsizlikni kuchaytirish va kontseptual ravishda jismoniy mavjud bo'lgandan ko'ra ko'proq xotiradan foydalanish imkoniyatiga ega bo'lishdir.

Virtual xotira DASD dagi faol RAM va harakatsiz xotirani birlashtirib, katta sonli tutashgan manzillarni hosil qiladi. Xotiraning virtualizatsiyasini virtual xotira tushunchasining umumlashtirilishi deb hisoblash mumkin.

Segmentlangan virtual xotira

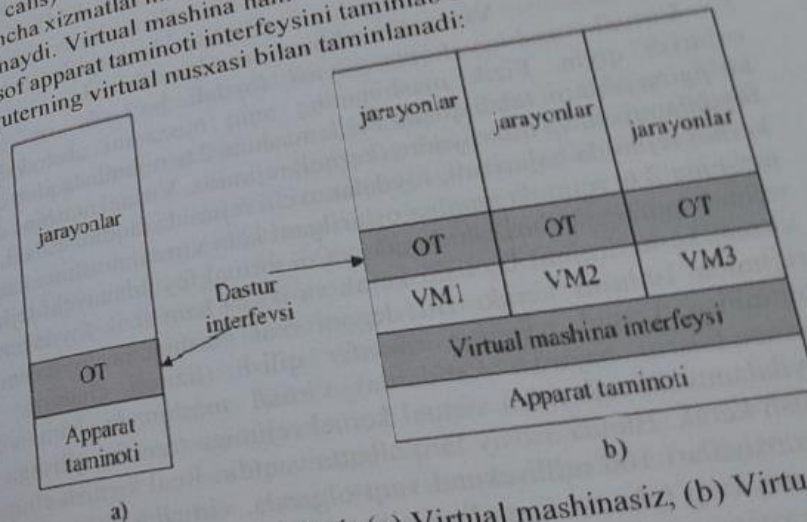
Ba'zi tizimlar, masalan Burroughs B5500, virtual manzil bo'shliqlarini o'zgaruvchan uzunlikdagi segmentlarga ajratib, disk raskadrovka o'rniga segmentatsiyadan foydalanadi. Bu yerda virtual manzil segment raqami va segment ichidagi ofsetdan iborat. Intel 80286 bir variant sifatida shunga o'xshash segmentatsiyasi sxemasini qo'llab-quvvatlaydi, lekin u kamdan-kam hollarda ishlatiladi. Har bir segmentni sahifalarga bo'lish orqali segmentatsiya va pagingdan birgalikda foydalanish mumkin; ushbu xotira tuzilishiga ega bo'lgan tizimlar, masalan, Multics va IBM System/38, odatda xotira himoyasini ta'minlovchi diskeshlashda ustunlik qiladi.

Disk xotira boshqaruvchisi (paging supervisor)

Operatsion tizimning ushbu qismi sahifalar jadvillarini yaratadi va boshqaradi. Agar apparat sahifada xatolik holatini keltirib chiqarsa, disk xotira boshlig'i ikkinchi darajali xotiraga kirib, sahifaning noto'g'ri ishlashiga olib kelgan virtual manzilga ega bo'lgan sahifani qaytaradi, sahifalar jadvillarini virtual manzilning jismoniy joylashishini aks ettirish uchun yangilaydi.

9.1. Virtual xotira konsepsiyasi

OT larni qurishda ko'p sathli yondoshuv o'zining mantiqiy yakunini virtual mashina konsepsiyasida topdi. Yuqorida takidlaganimizdek virtual mashina g'oyasi kompyuterning apparat ta'minotini bir nechta boshqa kompyuter sifatida abstraksiyalashdan iborat, bunda har bir hisoblash tizimi alohida mustaqil tizim sifatida bitta kompyuterda bir vaqtda ishlay oladi. Protessor vaqtini rejalashtirish va virtual xotira texnologiyalaridan foydalanib OT shunday muhit hosil qildiki bunda har bir jarayon vaqtini protessori va xotirada bajarilayotgandek holatni tasavvur qilish mumkin. Odatda jarayonlarda tizimli chaqiriq (sistemno'y vo'zov, system calls) va fayl tizimi (faylovoy sistema, file system) kabi qo'shimcha xizmatlar mavjudki, ular sof apparat taminoti orqali amalga oshirilmaydi. Virtual mashina ham bu funksiyalarni bajarmaydi, lekin aynan sof apparat taminoti interfeysini taminlab beradi. Har bir jarayon kompyuterning virtual nusxasi bilan taminlanadi:



9.1-rasm. Tizim modellari: (a) Virtual mashinasiz, (b) Virtual mashinali.

Virtual mashinani yaratishning bir necha sabablari bor bo'lib, asosan yagona apparat ta'minotni bir vaqtda ishlovchi bir necha bajariluvchi muhitlar (ya'ni OT lar) ga taqsimlab berishdir. Virtual

mashina bilan bog'liq qiyin muammolardan biri bu disk tizimidir. Tasavvur qiling, kompyuterda fizik jihatdan 3 ta qattiq disk mavjud, lekin 7 ta virtual mashina qattiq disk bilan ta'minlanishi kerak. Shuni aniqla, har bir virtual mashinaga bu disklar taqsimlanib bo'lmaydi, chunki virtual mashina dasturining o'zi ham virtual xotirada spuling (spooling)ni amalga oshirish uchun ma'lum xajmda diskdagi joyga muhtoj bo'ladi. Virtual disklar muammosi IBM virtual mashina operatsion tizimida minidisklar yordamida yechiladi. Har bir minidisk tizim tomonidan fizik disk hajmini kerakli darajadagi treklarga taqsimlash asosida tashkil qilinadi. Albatta bunda minidisklar hajmi yig'indisi fizik disklardagi mavjud xotira hajmidan ortiq bo'la olmaydi.

Virtual mashinaga ega bo'lgan foydalanuvchi kompyuterda ixtiyoriy operatsion tizimni yoki dasturlar paketini ishlatishi mumkin. IBM virtual mashina tizimi uchun foydalanuvchi CMS interaktiv operatsion tizimni ishlatishi mumkin. Virtual mashina dasturi fizik mashinadagi multidasturlash bilan bog'liqligini ham aytib o'tish lozim.

Virtual mashinani amalga oshirish

Virtual mashina kontsepsiyasi foydali bo'lsada, uni amalga oshirish qiyin. Fizik mashinaning aniq nusxasini abstraksiyalash ko'pgina ishlarni talab qiladi. Fizik mashina 2 ta rejimda taqdim etiladi: foydalanuvchi va mikroyadro (kernel) rejimida. Virtual mashina dasturi kernel rejimida bajariladi, foydalanuvchi rejimida taqdim etiladi. Fizik mashina 2 ta rejimda amalga oshirilgani kabi virtual mashina ham 2 ta rejimda amalga oshirilishi kerak, ya'ni virtual foydalanuvchi rejimi va virtual kernel rejimi bo'lishi kerak va ikkisi ham fizik foydalanuvchi rejimida ishlashi kerak. Bu degani real mashinada foydalanuvchi rejimidan kernel rejimiga transfer qilish (tizimli chaqiruv yoki komandalarni bajarish kabilarda) virtual mashinada ham virtual foydalanuvchi rejimidan virtual kernel rejimiga transfer qilishga olib kelish kerak. Bunda asosiy farq albatta vaqtdir. Real kiritish chiqarish operatsiyalari 100 millisekund vaqt olganda, virtual kiritish chiqarish kamroq vaqt (chunki u spuling qilinadi) yoki ko'proq vaqt (chunki u interpretatsiya qilinadi) olishi mumkin. Boshqa operatsion tizimlarda virtuallashitirish printspi 6 tarafdin markaziy protsessor bir qancha virtual mashinalar orasida taqsimlanishi (multidasurlanishi) virtual mashina ishini sekinlashuviga olib keladi.

Demak haqiqiy virtual mashinani amalga oshirishda barcha buyruqlar to'plamini hisobga olish lozimdir. Virtual mashina faqat IBM mashinalarida ishlaydi chunki virtual mashinadagi normal buyruqlar to'plami apparat taminot orqali bajariladi. Faqat intiyozli buyruqlargina (asosan kiritish-chiqarish buyruqlari) simulatsiyalanishi mumkin va shu sababdan ular sekinroq bajariladi.

Virtual mashina kontsepsiyasi bir qancha afzalliklarga ega. Shuni aytib o'tish lozimki, bu muhitda turli xil tizim resurslari to'liq himoyalangan bo'ladi, ya'ni har bir virtual mashina boshqa barcha virtual mashinalardan izolatsiyalangan bo'ladi, shuning uchun bu yerda himoyalani muammosi bo'lmaydi. Bir vaqtning o'zida ma'lumotlardan (fayllardan) tog'ridan to'g'ri ma'noda birga foydalanish yo'q. Ya'ni bunda foydalanish 2 usul bilan amalga oshiriladi. Birinchidan minidiskardan birga foydalanish mumkin, demak, undagi fayllardan ham, bu qoida fizik diskarni umumfoydalanuvchi (shared, obhiy dostup) statusga o'tqazgandek gap, lekin bu dasturiy jihatdan amalga oshiriladi. Ikkinchidan, virtual mashinalarni tarmoqda aniqlash mumkin, har bir virtual mashina virtual tarmoq orqali axborot almashishi mumkin.

Bu kabi virtual mashinalar operatsion tizimlarni tadqiq qilish, ularni qurish va sinash ushun xaqiqiy hisoblash mashinasidir. Odatda bir hisoblash mashinasida OT ni o'zgartirish qiyin ish. OT katta va kompleks dasturlar majmuidir va uning bir qismini o'zgartirish boshqa bir qismidagi xatoliklarga olib kelmasligini tasavvur qilish qiyin. Bu degani uni o'zgartirish kritik holatlarga olib keladi deganidir. Chunki OT kernel rejimida ishlaydi, bittagina ko'rsatgich (pointer, ukazatel) dagi xatolik butun boshli fayl tizimini zararlanishiga olib kelishi mumkin. Shuning uchun OT dagi barcha o'zgartirishlarni sinashni extiyotkorlik bilan amalga oshirish lozim. Bunda albatta tizimli daturlovchilarga virtual mashinadan foydalanish ko'pgina qulayliklarga olib keladi, ya'ni Operatsion tizimlarda virtuallashitirish printspi amalga oshiriladi. Chunki virtual mashinaning buzilishi real mashinaning buzilishiga olib kelmaydi. Virtual mashinani ixtiyoriy paytda almashitirish, qayta qurish mumkin.

OT hisoblash tizimida ishlaydi va uni nazorat qiladi. Shunday ekan OTdagi o'zgartirishlarni amalga oshirish va testlash yakunlanguncha ushbu tizimni to'xtatib turish zarur. Bu vaqt oralig'i odatda "tizim

rivojlanish davri" (system-development time) termini bilan yuritiladi. Shuning uchun bu davr odatda kech tunda yoki hafta so'ngida, ya'ni tizim yuklanish kam bo'lgan vaqtga rejalashtiriladi.

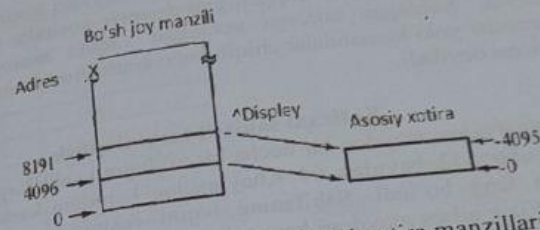
9.2. Virtual xotirani saxifali tashkil etish

Virtual xotirani amalga oshirishning bir necha usullari mavjud bo'lib bular virtual xotirani sahifali va segmentli tashkil etishdir.

Manchesterlik olimlar adres maydoni va xotira adresi tushunchalarini alohidalashni taklif qilishgan. Masalan, 16 bitli adres maydonli 4096 so'zli xotiraga ega mashinani qaraymiz. Bu kompyuterda ishlovchi dastur xotiraning 65 536 ta so'ziga murojaat qilishi mumkin (216q65 536). E'tibor bering: adreslanadigan so'zlar esa adres bitlari soniga bog'liq bo'lib, xotiradagi real so'zlar soniga bog'liq emas. Bu xotiraning adres maydoni 0, 1, 2, 3, ..., 65535 dan iborat. Xotira esa faqat 4096 so'zni saqlay oladi xolos.

Bu yerda 4095 gacha va 4096 dan keyingi adreslar orasida keskin farqlanish yuzaga kelgan. 4095 gacha foydali adres makoni, 4096 dan keyingilari foydasiz adres makoni deyilgan. Chunki 4096 dan keyingi adreslar xotiraning real adreslariga mos kelmagan. Adres maydoni va xotira adresi orasida farqlanish mavjud bo'lmagan, chunki ular orasida o'zaro bir xillik munosabati mavjud deb tushunilgan.

Adres maydoni va xotira adresini alohidalash g'oyasi quyidagicha tushuntiriladi. Har bir joriy vaqtda xotiraning 4096 ta so'ziga bevosita murojaat qilish mumkin. Bu esa ushbu so'zlar 0 dan 4095 gacha adresda joylashishi shart degani emas. Masalan, 4096-adresga murojaat qilinganda xotiraning 0-adresidagi so'z ishlatilishi, 4097-adresga murojaat qilinganda xotiraning 1-adresidagi so'z ishlatilishi, ..., 8191-adresga murojaat qilinganda xotiradagi 4095-so'z ishlatilishi lozim va h.k. Boshqacha aytganda adresli maydonni haqiqiy xotira maydonida akslanishi kuzatiladi.



9.2.-rasm. 4096 dan 8191 gacha virtual xotira manzillari, 0 dan 4095 gacha bo'lgan asosiy xotira manzillari.

Agar dastur 8192 adresdan 12287 adresga o'tishni bajarsa nima bo'ladi? Virtual xotirasiz mashinalarda xatolik sodir bo'ladi, ekranda "Mavjud bo'lmagan xotira adresi" yozuvi chiqadi. Virtual xotirali mashinada esa quyidagilar bajariladi:

- 1) 4096 dan 8191 gacha so'zlar diskda joylashadi
- 2) 8192 dan 12287 gacha so'zlar asosiy xotiraga yuklanadi.
- 3) Adreslarni akslanishi o'zgaradi. Endi 8192 dan 12287 gacha adreslar asosiy xotiradagi 0 dan 4095 gacha yacheykalarga moslanadi.
- 4) Dasturning bajarilishi davom etadi.

Adreslashni bunday avtomatik yoyish texnologiyasi xotirani sahifali tashkil qilinishi deyiladi, diskdan o'qiladigan dastur qismlari esa sahifalar deyiladi.

Adreslarni adres maydonidan xotiraning real adreslariga akslantirishining yana boshqa murakkabroq usuli ham mavjud. Dastur murojaat qilishi mumkin bo'lgan adreslarni virtual adres muhiti deyiladi, xotiradagi real adreslarni esa fizik adres muhiti deymiz. Xotirani taqsimlash sxemasi va sahifalar jadvali virtual adres va fizik adreslarni o'zaro moslashtiradi. Diskda to'liq virtual adres muhitini saqlash uchun joy yetarli deb hisoblanadi (juda bo'lmaganda joriy vaqtda bajarilib turgan qismini).

Dasturlar virtual adres muhitidan so'zlarni o'qishi va/yoki virtual adres muhitiga yozishi mumkin, aslida fizik xotira muhitida bunga joy bo'lmasa ham. Virtual xotira kompyuterining xotirasi katta ekanligi haqida tasavvur uyg'otadi xolos. Aslida kompyuterining fizik xotirasi

ancho kichik bo'ladi. Katta hajmli xotira tasavvuri xotiraning sahifli tashkil qilinishidan kelib chiqqanligini dasturiy vositalar bilan aniqlab bo'lmaydi. Xohlagan adresga xohlagan vaqtda murojaat qilinsa, ma'lumotlar yoki komandalar chiqib kelaveradi. Bu mexanizm shaffof mexanizm deyiladi.

Xotirani sahifali tashkil qilish

Virtual adres maydon bir necha bir xil hajmli sahiflarga bo'linadi. Odatda ular 512 baytdan 64 Kbaytgacha, kamdam-kam xollarda 4 Mbaytgacha teng bo'ladi. Sahifaning hajmi ikkining darajasiga teng bo'ladi. Fizik adres maydoni ham teng miqdordagi qismlarga shunday bo'linadiki, har bir qism bitta sahifani sig'dira oladi. Asosiy xotiraning bu qismlari sahifali kadrlar deyiladi.

Bu yerda asosiy xotira faqat bitta sahifali kadri saqlaydi.

Adres bayt ham, so'z ham bo'lishi mumkin, bunda ketma-ket so'zlar ketma-ket adreslarni bildiradi. Rasmdagi virtual xotirani sahifalar jadvali yordamida tashkil qilish mumkin. Jadval elementlari soni va virtual xotira sahifalari soni teng bo'ladi. (b) da 4 K hajmli 8 ta sahifalik fizik xotira keltirilgan.

32 bitli virtual adresni asosiy xotiraning fizik adreslarida qanday ifodalash mumkin? Ma'lumki, fizik xotira faqat real adreslar bilan ishlaydi, virtual adreslarni qabul qilmaydi. Buning uchun har bir kompyuterda virtual adreslarni fizik adresga aylantirib beruvchi qurilma mavjud bo'ladi. Bu qurilma xotirani boshqarish kontrolleri (MMU – Memory Management Unit) deyiladi. U bevosita markaziy protsessor mikrosxemasida yoki alohida joylashishi mumkin. Bizning misolda MMU 32 bitli virtual adresni 15 bitli fizik adresga aylantiradi, shu sababli unga 32 bitli kiruvchi registr va 15 bitli chiquvchi registr kerak bo'ladi.

Virtual manzil sahifasi

15	E144C P 65535
14	57344 P 61439
13	53248 P 57343
12	49152 P 53247
11	45056 P 49151
10	40960 P 45055
9	36864 P 40959
8	32768 P 36863
7	28672 P 32767
6	24576 P 28671
5	20480 P 24575
4	16384 P 20479
3	12288 P 16383
2	8192 P 12287
1	4096 P 8191
0	0 P 4095

a

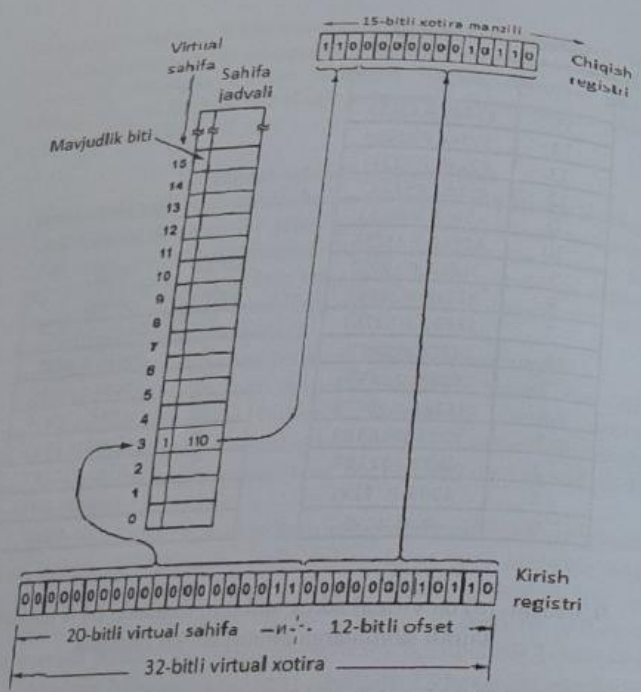
Pastki 32K asosiy xotira manzillari

Sahifa kadri	Fizik adres
7	28672 P 32767
6	24576 P 28671
5	20480 P 24575
4	16384 P 20479
3	12288 P 16383
2	8192 P 12287
1	4096 P 8191
0	0 P 4095

b

9.3-rasm. (a) da virtual adres muhitini dastlabki 64 K hajmini 4 K hajmli sahifalarga ajratilgani ko'rsatilgan.

Dastlab MMU kerakli sahifa hozirgi vaqtda xotirada mavjudligini aniqlaydi. Hozirgi vaqtda 220 ta virtual sahifalar va 8 ta sahifali kadr bor ekan, hamma virtual sahifalar bir vaqtda xotirada mavjud bo'la olmaydi. MMU sahifalar jadvali elementining mavjudlik bitini tekshiradi. Ushbu holatda bu 1 ga teng va sahifa xotirada mavjudligini bildiradi. Keyin tanlangan elementdan sahifa kadri qiymatini olish lozim (hozirgi holatda – 6) va uning chiquvchi 15 bitli registrining yuqori 3 bitini nusxalash kerak. Aynan 3 ta bit shartligi, fizik xotirada 8 ta sahifa kadri borligidan kelib chiqadi. Bu operatsiya bilan parallel ravishda virtual adresning quyi 12 biti (sahifadagi ko'chirish maydoni) chiquvchi registrning quyi 12 bitiga nusxalanadi. Shu tartibda hosil qilingan 15 bitli adres keshga yoki asosiy xotiraga yuboriladi.



9.4-rasm. Virtual xotira manzilidan asosiy xotira manzilini shakllantirish

9.3. FIFO, LRU va "ikkinchi imkoniyat" algoritmlari

FIFO (First Input First Output) bu avval yozilgan (ya'ni, kirgan) ma'lumotlar undan so'ng yangi ma'lumotlar kiritilganda old tomonga siljishini nazarda tutuvchi xotirani tashkillashtirish usulidir. Odatda bunday tashkil etilgan xotira ma'lumotlarni qabul qilish-uzatish paytida bufer xotirasi sifatida qo'llaniladi.

Queue ham Stack singari *chiziqli ma'lumot tuzilmasi* bo'lib, hayotdagi oddiy navbat singari ishlaydi. Shu sababli Stackdan farqli

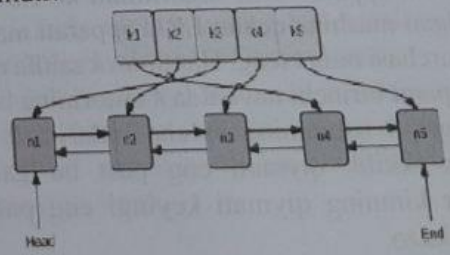
o'laroq Queue da eng oxirgi qo'shilgan elementga emas birinchi qo'shilgan elementga birinchi bo'lib "xizmat ko'rsatiladi". Operatsiyalarni bunday ko'rinishda amalga oshirilishi esa FIFO deb ataladi. Queue ni tasavvur etish uchun quyidagi rasmni ko'rib chiqamiz.



9.5-rasm. Queue ni ishlashi tasavvur etish

Queue ga dasturiy misollar sifatida printerga chop qilishni uzatishni, yoki protsessor operatsiyalarni amalga oshirish jarayonini misol keltirish mumkin (protsessor ishlashi har doim ham FIFO ga asoslanmaydi).

LRU (Least Recently Used) – anchadan buyon ishlatilmayotgan elementlar yangilari tomonidan siqib chiqarilayotgan keshlashni boshqarish algoritmidir.



9.6-rasm. Yaqinda ishlatilgan (LRU) sahifani almashtirish algoritmi

Optimal algoritimga yaxshi yaqinlashish, so'nggi bir necha ko'rsatmalarda juda ko'p ishlatilgan sahifalar, ehtimol keyingi bir nechta yana ko'p ishlatilishini kuzatishga asoslangan. Aksincha, asrlar davomida ishlatilmagan sahifalar, ehtimol, uzoq vaqt davomida ishlatilmay qolishi mumkin. Ushbu g'oya amalga oshiriladigan algoritmi taklif qiladi: sahifada nosozlik yuz berganda, uzoq vaqt davomida ishlatilmaydigan sahifani tashlang. Ushbu **strategiya LRU (yaqinda ishlatilgan eng yaqinda)** paging deb nomlanadi.

LRU nazariy jihatdan amalga oshirilsada, bu arzon emas. LRUni to'liq amalga oshirish uchun xotirada barcha sahifalarning bog'langan ro'yxatini, eng so'nggi ishlatilgan sahifani va eng oxirgi ishlatilgan sahifani orqada saqlash kerak. Qiyinchilik shundaki, ro'yxat har bir ma'lumot uchun yangilanishi kerak. Ro'yxatdagi sahifani topish, uni o'chirish va keyin oldinga siljitish, hatto apparatda ham juda ko'p vaqt sarflaydigan operatsiya (agar bunday uskunani qurish mumkin deb hisoblasak).

Biroq, LRUni maxsus apparat vositalari bilan amalga oshirishning boshqa usullari mavjud. Avvalo eng oddiy usulni ko'rib chiqaylik. Ushbu usul qo'shimcha qurilmani har bir ko'rsatmadan keyin avtomatik ravishda ko'paytiriladigan 64 bitli hisoblagich C bilan jihozlashni talab qiladi. Bundan tashqari, har bir jadval jadvalidagi yozuvda hisoblagichni o'z ichiga oladigan darajada maydon bo'lishi kerak. Har bir xotira ma'lumotidan so'ng, C ning joriy qiymati havola qilingan sahifa uchun jadvallar jadvalida saqlanadi. Sahifada xato bo'lsa, operatsion tizim eng pastini topish uchun sahifalar jadvalidagi barcha hisoblagichlarni tekshiradi. Ushbu sahifa eng kam ishlatilgan sahifadir.

Endi ikkinchi apparat LRU algoritmini ko'rib chiqamiz. N sahifa ramkalari bo'lgan mashina uchun LRU apparati matritsani $n \times n$ o'tkir; n bit, dastlab barchasi nolga teng. Har doim k sahifa ramkasiga murojaat qilinganida, apparat birinchi navbatda k qatorining barcha bitlarini 1 ga o'rnatadi, so'ngra k ustunining barcha bitlarini 0 ga o'rnatadi. Har qanday lahzada ikkilik qiymati eng past bo'lgan satr eng yaqin ishlatilgan qator kimning qiymati keyingi eng past bo'lsa, yaqinda ishlatilgan va hokazo.

O'qish-yozish qo'shimcha almashinuvlarsiz amalga oshiriladi, chunki faylni ochishda biz berilganlarni joylashtirish diapazoniga ega bo'lamiz. O'qish ushbu struktura blokiga mos ravishda amalga oshiriladi va xech qanday qo'shimcha ma'lumot talab etilmaydi, almashinuv ham mos ravishda tezda amalga oshiriladi. Endi qarab chiqaylik, bunday faylga qo'shimcha ma'lumot talab etilmaydi, bo'sh fazo joy yo'q. Bu xolatda tizim ikki xil yo'l tutishi mumkin. Birinchidan, u bizga joy yo'qligini aytadi va biz o'zimiz nimadir qilishimiz kerak bo'ladi, ya'ni qandaydir ushbu faylni biror joyga ko'chirib turadigan va kerakli ma'lumotni qo'shadigan jarayonni qo'yamiz. Bunday ko'chirish yetarli darajada qimmatga jarayonni funktsiya hisoblanadi. Bunda ikkinchi imkoniyat - almashinuvni rad etiladi. Bu esa faylni ochish jarayonida avvaldan qo'shimcha joy olib qo'yish kerakligini anglatadi; bu xolda fayl tizimi bufering bo'sh o'lchamini tekshiradi va u kam bo'lsa, ushbu faylni joylashtirish uchun bo'sh joy qidiradi.

Ikkinchi imkoniyatli sahifani almashtirish algoritmi deb nomlanuvchi FIFO sahifasini almashtirish algoritmining o'zgartirilgan shaklidir. U FIFO kabi navbatning old tomoniga qarab ishlaydi, lekin darhol ushbu sahifani disk raskadrovka qilish o'rniga, unga havola qilingan bit o'rnatilganligini tekshiradi. Agar u o'rnatilmagan bo'lsa, sahifa almashtiriladi. Aks holda, havola qilingan bit o'chiriladi, sahifa navbatning orqa qismiga kiritiladi (xuddi yangi sahifaday) va bu jarayon takrorlanadi. Buni dumaloq navbat deb hisoblash ham mumkin. Agar barcha sahifalar havola qilingan bit to'plamiga ega bo'lsa, ro'yxatdagi birinchi sahifaning ikkinchi uchrashuvida ushbu sahifa almashtiriladi, chunki u endi havola qilingan bitni o'chirib tashlagan. Agar barcha sahifalarda mos yozuvlar biti tozalangan bo'lsa, unda ikkinchi imkoniyat algoritmi sof FIFOga aylanadi.

Nomidan ko'rinib turibdiki, Ikkinchi imkoniyat har bir sahifaga "ikkinchi imkoniyat" beradi - havola qilingan eski sahifa, ehtimol ishlatilayotgan bo'lishi mumkin va havola qilinmagan yangi sahifa bilan almashtirilmasligi kerak.

9.4. Nazorat savollari

1. Virtual xotira nima va uning ishlash prinsipi?
2. Virtual xotira konsepsiyasidan iborat?
3. Virtual xotirani saxifali tashkil etish usullari?
4. FIFO, LRU va "ikkinchi imkoniyat" algoritmlarini tushuntirib bering?

10. OPERATSION TIZIMLARDA FAYL TIZIMLARI.

10.1. Fayl tizimi funksiyalari va ma'lumotlar ierarxiyasi

Fayl deganda, odatda nomlangan, bir xil tuzilishga ega bo'lgan yozuvlardan tashkil topgan ma'lumotlar to'plami tushuniladi. Bu ma'lumotlarni boshqarish uchun, mos ravishda fayl tizimlari yaratiladi. Fayl tizimi, ma'lumotlar mantiqiy strukturasi bilan ish olib borish imkonini beradi. Aynan fayl tizimi, ma'lumotlarni disklarda yoki biror-bir boshqa jamlamada tashkil etish usulini aniqlaydi. Fayl tizimining qabul qilingan spetsifikatsiyalari bo'yicha, fayllar bilan ishlashni amalga oshiruvchi maxsus tizimli dasturiy ta'minotni, ko'pincha fayllarni boshqaruv tizimi deyiladi. Aynan, fayllarni boshqarish tizimi, fayl ma'lumotlarini yaratish, yo'qotish tashkil etish, o'qish, yozish, modifikatsiya qilish va joyini o'zgartirish va shu bilan birga fayllarga murojaatni va fayllar tomonidan foydalaniladigan resurslarni boshqarishga javob beradi.

Fayllarni boshqarish tizimining "FBT" asosiy vazifasi, bizga kerakli bo'lgan yozuvning aniq fizik adressini ko'rsatib, quyi darajada murojaat o'rniga, fayl ko'rinishida tashkil etilgan ma'lumotlarga murojaatning qulay usulini berishidir, ya'ni fayl nomi va undagi yozuvni nomini ko'rsatib, mantiqan murojaat qilishga imkon berishdir.

Fayllarni boshqarish tizimi yordamida, foydalanuvchilarga quyidagi imkoniyatlar yaratiladi:

- foydalanuvchilarning muloqat funksiyalarini (uning ma'lumotlari bilan) amalga oshiruvchi va fayllarni boshqarish tizimidan faol foydalanuvchi, maxsus boshqaruvchi funksiyalar yordamida yoki o'z dasturlaridan, nomli ma'lumotlar to'plamini (fayllarni) yaratish, olib tashlash va qayta nomlash (va boshqa operatsiyalar);
- disksiz periferik qurilmalar bilan fayl kabi ishlash;
- fayllar orasida, qurilmalar o'rtasida (va teskari) ma'lumotlar almashinish;
- fayllarni boshqarish tizimi dasturiy modullariga murojaat usuli bilan fayllar bilan ishlash (API ning bir qismi fayllar bilan ishlashga mo'ljallangan);
- fayllarni xuquqsiz murojaatdan himoya qilish.

Qoida bo'yicha hamma zamonaviy OT lar o'z fayllarini boshqarish tizimlariga egadir. Ba'zi OT lar esa, bir nechta fayl tizimlari bilan ishlash imkoniga egadirlar (bir nechta ichidan bittasi yoki bir nechta bilan bir vaqtda). Bu hollarda, montirovka qilinadigan fayl tizimlari to'g'risida so'z boradi (montirovka qilinadigan fayllarni boshqarish tizimlarini qo'shimcha sifatida o'rnatish mumkin) va bu borada ular mustasnodir.

Shu narsa ma'lumki, fayllarni boshqaruv tizimi, OT ning asosiy komponentasi bo'lgan holda, undan mustaqil emas, chunki API ning mos chaqiriqlaridan faol foydalanadi. Shu bilan birga, FBT ning o'zi API ni yangi chaqiriqlar bilan boyitadi.

Demak, shuni ta'kidlash mumkinki, ixtiyoriy FBT o'z-o'ziga mavjud emas, balki ular aniq OT muhitida ishlash uchun yaratiladi. Misol uchun hammaga topish fayl tizimi FAT ni keltirish mumkin (File Allocation Table – fayllarni joylashtirish jadvali). Bu tizim, fayllarni boshqarish tizimi sifatida ko'p hollarda ishlatiladi.

Bu fayl tizimining turli OT larda ishlaydigan, turli versiyalari mavjuddir: FAT, FAT 12, FAT 16, super FAT va x.k.lar.

Bu FBT ning MS DOS OT dan Windows NT bilan ishlaydigan versiyalari mavjud. Boshqacha qilib aytganda, ma'lum fayllar tizimiga mos tashkil etilgan fayllar bilan ishlashda, har bir OT uchun o'z fayllarni boshqarish tizimi ishlab chiqishi shart. Va bu fayl tizimi, qaysi OT uchun mo'ljallangan bo'lsa, o'sha OT da ishlaydi, ammo boshqa fayllarni boshqaruv tizimi yordamida yaratilgan fayllarga murojaat imkonini ta'minlaydi (boshqa OT FBT bu OT FBT printsiplari asosida ishlaydigan bo'lishi kerak).

Fayl tizimi – bu ma'lumotlarning qabul qilingan sertifikatsiyalari bo'yicha tashkil etilgan to'plamlari majmuasi bo'lib, ular fayllarga murojaat uchun zarur bo'lgan adresli ma'lumotlarni olish usulini aniqlaydi.

Demak, fayl tizimi termini, avvalambor fayllarga tashkil etilgan ma'lumotlarga murojaat prinsiplarini aniqlaydi. Xuddi shu terminni u yoki bu jamlamada joylashgan aniq fayllarga nisbatan ham ishlatish mumkin. Fayllarni boshqarish termini esa, fayl tizimining aniq realizatsiyasiga (amalga oshishiga) nisbatan ishlatish mumkin, ya'ni bu tizim – aniq OT larda fayllar bilan ishlashni ta'minlaydigan dastur modullari to'plamidir.

Inson ishlaydigan ma'lumotlar, odatda strukturalashtirilgandir. Bu avvalo, ma'lumotlarni samarali saqlashni tashkil etishga imkon beradi, qidiruvni osonlashtiradi, nomlashda qo'shimcha imkoniyatlar yaratadi. Xuddi shuningdek, fayllar bilan ishlashda ham iloji boricha strukturalashtirish mexanizmini kiritish kerak. Hammadan osoni ierarxik munosabatlarni tashkil etishdir. Buning uchun, katalog etilgan ma'lumotlar to'g'risidagi ma'lumotni o'z ichiga oladi.

Boshqacha aytganda, katalogda fayllar diskriptori saqlanadi. Agar fayllar, blok qurilmasida tashkil etilgan bo'lsa, aynan katalog yordamida FBT, shu ma'lumotlar joylashgan blok adreslarini topadi. Katalog, maxsus tizimli ma'lumot strukturasi emas, (asosiy, ildiz katalog deyiladi), fayl o'zi ham bo'lishi mumkin. Bunday fayl orasida maxsus tizimli qiymatga ega bo'lib, uni boshqa oddiy fayllar orasida ajratish zarurdir. Fayl Katalog ko'pincha "podkatalog" deb ataladi. Agar fayl katalog, boshqa fayllar haqida ma'lumotni o'zida saqlasa, ular orasida ham o'z navbatida fayl-kataloglar bo'lsa, biz xech qanday chegaralanmagan ierarxiyaga ega bo'lamiz.

Undan tashqari, bunday fayl ob'ektlarni fayl katalog kiritish, fayl tizimini nafaqat strukturalash, ildiz katalogdagi elementlar sonini chegaralanmagan muammosini hal qiladi. Fayl katalogida elementlar soniga chegara yo'q, shuning uchun katta o'lchamdagi kataloglarni yaratish mumkin.

10.2. Fayllarni joylashtirish jadvali

Fayllarni joylashtirish jadvali, juda muhim ma'lumotlar strukturasi. Aytish mumkinki, u, ma'lumotlar sohasining holati va uning u yoki bu fayl ob'ektiga mutanosibli tavsiflanadigan ma'lumotlar sohasi adres kartasidan iborat.

Ma'lumotlar sohasi, klasterlarga bo'linadi. Klaster bu- mantiqiy disk adres makonida bir yoki bir nechta ayqash sektorlardan iboratdir (aniqrog'i-faqat ma'lumot sohasida). Klaster, faylga ajratiladigan, xotira diskning adreslanadigan minimal birligidir. Klasterlar, mantiqiy diskning ma'lumotlar sohasidagi adreslanadigan birligi sonini kamaytirish uchun kiritilgandir.

10.3. Fayl tizimlari imkoniyatlari va ishonchligi

Fayl tizimi bu ma'lumotlarni strukturasi va metodlaridir. Ular operatsion tizim tomonidan fayllarni olishda yoki disk qismida saqlash uchun qo'llaniladi. Ko'pincha fayl tizim deganda biz diskning bo'lagi yoki disk ma'lumotlarini aytamiz. Chunki bular faylni saqlash uchun qo'llaniladi. Lekin fayl tizimning va disk yoki disk bo'lagining farqini bilish kerak. Masalar ba'zi dasturlar diskga murojaat qilishda to'liq ruxsat beradi. Agarda shu vaziyatda diskda fayl tizimi mavjud bo'lsa fayl tizimi buzilgan bo'lishi mumkin.

Diskni fayl tizimi sifatida qo'llashdan oldin fayl tizimining o'zi inisializatsiya amaladan o'tishi kerak. Ushbu harakat fayl tizimining yaratilishi deb aytiladi. Fayllar joylashish jadvali o'zi fayl emas. Ushbu jadval DOS yoki BIOS maxsus funksiyalari bilan o'qib olinishi mumkin.

Fayl tizimida katalog bu fayl boshqa fayllar nomlarini ro'yhatdan o'tkazish va ro'yhatdan o'tgan nomlar bo'yicha ruxsatnoma ta'minlovchi fayl. Katalog fayllarning guruhini aniqlovchi ma'lumotlarni saqlaydi. Har bir diskda bo'sh katalog mavjud. Ushbu katalog nomi joylashgan diskning nomi bilan bir deb hisoblasak bo'ladi. Shu sababli fayl tizimida ko'p pog'onali daraxtga o'xshash katalog strukturasi mavjud. Tashqi fayllar uchun 2 tur ruxsatnoma mavjud. Ketma – ket va to'g'ri.

Ketma – ket ruxsatnomada fayl qiymatlari ko'rib chiqiladi. Ushbu qiymatlar qabul qilish navbatda uzatiladi. Agarda fayl ochiq bo'lsa uzatish amali faylning boshidan boshlanadi. To'g'ri ruxsatnomada fayl elementlarining to'planmasi aniq joylarni egallab turadigan chiziqli tartibda qarab chiqiladi. Ba'zi qiymati faylning elementiga yoki elementidan uzatilishi mumkin. Xohlagan holatda elementning joylashuvi elementning indeksi bilan aniqlanadi.

Fayllarni boshqarish tizimlari, ko'pgina zamonaviy OT larning asosini tashkil etadi. M-n, UNIX OT, fayl tizimisiz ishlamaydi, unda fayl tizimi asosiy tushunchalardan biridir. Hamma zamonaviy OT lar fayllardan va ular bilan ishlash uchun mos dastur ta'minotidan foydalanadi. Gap shundaki, birinchidan, fayl tizimi orqali ma'lumotlar bo'yicha ko'pgina ishlov beruvchi dasturlar baholanadi. Ikkinchidan, bu tizim orqali disk makonini markazlashtirilgan holda taqsimlash va

ma'lumotlarni boshqarish muammolari yechiladi. Va nihoyat, foydalanuvchilar o'z ma'lumotlariga murojaat qilishning oson usullariga ega bo'ladilar, bu ma'lumotlarni, to'g'ri xotira qurilmalarida joylashtiradilar.

Turli OT lar va Tashqi xotiraning turli qurilmalari uchun yaratilgan ko'pgina fayl tizimlari mavjuddir. Ularda, mos ravishda ma'lumotlarni joylama (nositel) da joylashtirishning har xil prinsiplaridan foydalaniladi. Biz, FAT, FAT 32 va NTFS fayl tizimlari bilan tanishamiz. Ayniqsa, hozirgi kunda eng ko'p tarqalgan fayl tizimi bilan – NTFS bilan tanishish muhim ahamiyatga egadir.

Fayllarning belgisi aloqa bu maxsus fayl. Ushbu fayl boshqa faylga yo'nalishini o'zida saqlaydi. Berilgan katalogning elementi belgisi aloqaga tegishli haqida ma'lumot indeksining deskriptorida yoziladi. Odatda oddiy buyruqlar faylga ruxsatnoma qilishi mumkin. Ushbu tizim fayl fazasini taqsimlashni yaxshi tashkil qiladi.

Har xil fayl maxsus bir tupga tegishli. Odatda fayllarni bir nechta turga ajratamiz.

Fayl tizimlarining taqsimlanishi

Har bir taqsimlangan tizimning asosiy komponenti bu fayl tizimi taqsimlangan ham markaziy yo'naltirilgan tizimlarda ham fayl tizimining asosiy maqsadi dasturlarni va ma'lumotlarni saqlab olish va ularga ruxsatnoma berish. Fayl tizimi bitta yoki bir nechta qurilma bilan ta'minlanishi mumkin. Ushbu qurilmalar fayl server deb aytiladi. Fayl serverlar o'qishga yoki yozishga so'roqlarni qabul qilib natijalarni so'rovchi server emas kompyuterlarga qaytarib beradilar.

Fayl serverlar odatda ierarxik tizimga ega. Har bir fayl tizimi o'ziga ildiz katalogga ega ham past pog'onali katalogga ega. Server va klient ushbu fayl tizimlari o'zining lokal fayl tizimlarga ulay oladi. Ushbu ulanadigan fayllar serverda saqlanib qoladi.

10.4. FAT, NTFS, HPES, ext2, ext3, ext4 va boshqa fayl tizimlari.

10.4.1. FAT, VFAT va FAT32 fayl tizimlari

FAT fayl tizimi, o'z nomiga quyidagi ma'lumotlarni o'z ichiga olgan oddiy jadval orqali ega bo'lgan:

- fayl yoki uning fragmentlari uchun ajratilgan, mantiqiy diskning bevosita adresslanuvchi qismlari (uchastkalari);
- disk makoni bo'sh sohalari;
- diskning defektlil sohalari (bu sohalarda defekt joylariga ega bo'lib, ma'lumotlarni o'qish va yozishni xatosiz bajarishga kafolat bermaydi).

FAT fayl tizimida, ixtiyoriy mantiqiy disk ikki sohaga: tizimli soha va ma'lumotlar sohasiga bo'linadi.

Mantiqiy disk tizimli sohasi, formatlash vaqtida initsiallashtiriladi, keyinchalik esa, fayl strukturasi bilan ishlaganda yangilanadi. Mantiqiy disk, ma'lumotlar sohasi oddiy fayl va fayl kataloglarni o'z ichiga oladi; bu ob'ektlar ierarxiyani tashkil etadi, bu ob'ektlar ildiz katalogga bo'ysunadi.

Katalog elementi fayl ob'ektini tavsiflaydi, u oddiy fayl yoki fayl-katalog bo'lishi mumkin. Ma'lumotlar sohasiga, tizimli sohadan farqli ravishda, OT ning foydalanuvchi interfeysi orqali murojaat qilinadi. Tizimli soha quyidagi tashkil etuvchilardan iboratdir (mantiqiy adress sohasida ketma-ket joylashgan):

- yuklanish yozuvi (Boot Record, BR);
- rezervlangan sektorlar (Reserved. Sectors, Res.Sec);
- fayllarni joylashtirish jadvali (FAT);
- ildiz katalog (Root Directory, R Dir).

NFS bu eng taniqli Unix va Linux tizimlari uchun birgalikda fayllarga ruxsatnoma olish mexanizmi

FAT hamma MS DOS versiyalarida qo'llanilgan operatsion tizim hamda OS/2 versiyasida ham qo'llangan. FAT tizimining har bir mantiqiy diski o'zining FATga ega. Ushbu jadval 2 ta vazifani bajaradi.

1. Mantiqiy diskda joylashgan har bir faylni taqsimlanish axborotni saqlaydi. Taqsimlanish modullar ro'yxatning shaklida
2. Qaysi taqsimlanish modullar bo'sh ekanligini ko'rsatadi

FAT boshlang'ich fayl tizimi muhim xarakteristikalaridan biri fayl nomlari 8.3 formatidan foydalanishdir. FAT standart tizimiga (FAT 16 ko'zda tutilmoqda) yana 2 ta, keng tarqalgan MS OT – Windows 95 va Windows NT uchun ko'rinishlari: V FAT (FAT virtual tizimi) va FAT 32 tizimidir. Hozirgi vaqtda FAT 32 – Windows Millennium Edition, Windows 2000 va Windows XP tomonidan qo'llaniladi, bu FAT 32, tizimining Windows NT va Linux uchun ham ishlab chiqilgan versiyalari mavjuddir.

FAT va VFAT fayl tizimlari asosiy kamchiligi, katta o'lchamli klasterlardagi yo'qotishlar va mantiqiy o'lchamiga bo'lgan chegaralardir. Shuning uchun ham MS Win 95 OEM Service Release 2 uchun FAT 32 keldi. U to'liq mustaqil 32 razryadli fayl tizimidir va u oldingi versiyalarga nisbatan ko'p mukammalliklarga egadir. Eng asosiysi, FAT 32 disk sohasini samarali sarflaydi.

FAT16 fayl tizimining qo'llash uchun odatda qo'shimcha 200MB bo'sh joy kerak bo'ladi. Bunga sabab klasterlarning kattaligi agarda diskning bo'limi 26baytdan ko'p bo'lsa FAT16 disklarda qo'llaniladi. FAT16 ning kamchiliklari: Katalogi 512 ta yozuv bilan cheklangan va faylning nomlari uzun. FAT16 65 536 registr bilan cheklangan. Klasterlar rezervga tushgani sababli 65 524. Eng katta FAT16 disklar WINDOWS 2000 klasterlari

10.4.2. HPFS fayl tizimi.

HPFS fayl tizimi (High File System – yuqori unumdorlikka ega bo'lgan fayl tizimidir) birinchi marta OSG'2 va Law Manager OT larida paydo bo'ldi. Bu fayl tizimi, IBM va MS kompaniya mutaxassislari tomonidan MVS, VM/EMS fayl tizimlari va virtual murojaat usuli tajribasi asosida ishlab chiqildi. HPFS ko'pmasalalik rejimi fayl tizimi sifatida yaratila boshladi va katta o'lchamli disklardagi fayllar bilan ishlashda yuqori unumdorlikni ta'minlash uchun mo'ljallangan edi.

HPFS asosi qilib olingan fayllarni diskda joylashtirish prinsipi, fayl tizimini unumdorligini, va uning ishonchliligi va buzilishlar qat'iyliligini oshiradi.

HPFS fayl tizimi, FAT bilan taqqoslaganda quyidagi ustunliklarga egadir:

- yuqori unumdorlik;
 - ishonchlilik;
 - fayl va kataloglarga murojaatni moslanuvchi holda bosh imkonini beradigan kengaytirilgan atributlarni qo'llash;
 - disk makonidan samarali foydalanish.
- Bu ustunliklar HPFS strukturasiidan kelib chiqadi.

10.4.3. NTFS fayl tizimi

NTFS (New Technology File System – yangi texnologiya fayl tizimi) fayl tizimi nomida yangi so'zi mavjuddir. Haqiqatda, NTFS fayl tizimi, taniqli FAT 16 (va hatto FAT 32)ga nisbatan sezilarli mukammalliklar va o'zgarishlarni o'z ichiga olgan. Foydalanuvchi nuqtai-nazaridan qaraganda, fayllar har doimdagidek (oldindagidek), Windows muhitida ishlash vaqtida ko'pincha "papka" deb ataluvchi kataloglarda saqlanadi. Ammo unda talay yangi xususiyat va imkoniyatlar paydo bo'ldi.

NTFS fayl tizimi asosiy imkoniyatlari. NTFS ni loyihalashda alohida diqqatni ishonchlikka, katalog va fayllarga murojaatni chegaralash mexanizmiga, kengaytirilgan funktsionallikka, katta xajmdagi disklarni qo'llashga va h.k.larga qaratildi. Bu tizim OSG-2 V.3 doirasida ishlab chiqila boshladi, shuning uchun ham u HPFS fayl tizimi ko'pgina qiziqarli xususiyatlarini olgan.

Ishonchlik. Yuqori unumdorlikka ega bo'lishi kerak. Bu esa foydalanish tizimlari, yuqori ishonchlikka ega bo'lishi kerak. Bu esa NTFS tizimining eng muhim elementidan iboratdir. NTFS tizimi, o'z-o'zini tiklashning ma'lum vositalariga ega. Bu vositalar tizim yaxlitligi (butunligi) ni tekshiradigan, ya'ni tranzaktsiya jurnalini olib boradigan turli mexanizmlarni o'z ichiga oladi.

NTFS, qayta yuklanmasdan tinimsiz ishlashni ta'minlashi muhim bo'lgan server fayl tizimi sifatida ishlab chiqilgani uchun, unda xuddi HPFS dagi kabi, defektlar sektorlarni avariya almashtirish mexanizmi, ishonchlikni oshirish uchun kiritilgan. Boshqacha aytganda, ma'lumotlarni o'qishda tizim rad etsa, u holda fayl tizimi bu ma'lumotlarni o'qishga va bu maqsad uchun disk makonini maxsus rezervlar qayta ko'chiradi, defekt joyini-sektorni belgilab, unga boshqa murojaat qilmaydi.

Fayl va kataloglarga murojaat chegaralari. NTFS fayl tizimi Windows NT OT i xavfsizlik ob'ekt modelini qo'llaydi va hamma tom, katalog va fayllarga mustaqil ob'ekt sifatida qaraydi. NTFS tizimi xavfsizlikni fayl va kataloglar darajasida ta'minlaydi. Bu degani, tom, katalog va fayllarga murojaat xuquqi, foydalanuvchi hisob yozuvi va u mansub bo'lgan guruhga bog'liqdir. Foydalanuvchi har gal fayl tizimi ob'ektiga murojaat qilganida uning xuquqi ob'ektning murojaatini

boshqarish ro'yxatidan tekshiriladi (ACL). Agar foydalanuvchi zarur xuquqqa ega bo'lsa, uning so'rovnomasi qondiriladi, aks holda so'rovnoma rad etiladi.

Kengaytirilgan funktsionallik. NTFS tizimi mumkin bo'lgan qo'shimcha imkoniyatlar aks ettirilgan loyihalashtirilgan. Unda ko'pgina ma'lumotlar hisobga olinib loyihalashtirilgan. Unda ko'pgina (foydalanuvchi belgilaydigan).

Windows 2000/XP tizimlarida, NTFS fayl tizimidan foydalanishda kvotilashtirishni kiritish mumkin, bu foydalanuvchilar o'z fayllarini disk makonining ularga ajratilgan kvotasi chegarasida saqlashi mumkin.

NTFS tizimi katta disklar bilan ishlashni hisobga olib yaratilgan chegaralash uchun berilgan fayl va kataloglarning maxsus kengaytirilgan to'plami. Ular faqat NTFS fayl tizimi o'rnatilgan to'plamlarda mavjuddir. Ruxsatlar moslanuvchan himoyani ta'minlaydi, chunki ularni ham katalog, ham alohida fayllarga qo'llash mumkin; ular lokal foydalanuvchilarga (himoyalangan papka va fayllar bo'lgan kompyuterlarda ishlaydigan) ham, Tarmoq orqali resurslarga ulanadigan foydalanuvchilarga ham mansubdir. Ruxsat va xuquqlarni chalkashtirib bo'lmaydi. Bular umuman har xil tushunchalardir.

NTFS ruxsatlari, birinchi navbatda, resurslarni shu resurs joylashgan kompyuterda ishlayotgan foydalanuvchidan himoya uchun ishlatiladi. Ammo ulardan, masofadagi umumiy papkaga Tarmoq orqali murojaat qiluvchi foydalanuvchilarga ham ishlatish mumkin.

NTFS ruxsatlar himoyani yuqori darajada tanlashni ta'minlaydi: har bir papkadagi faylga o'z ruxsatini o'rnatish mumkin. har bir foydalanuvchining ruxsatlari, ma'lum amallar bo'yicha farqlanadi. Demak, har bir fayl ob'ekti o'zining murojaatini boshqarish ro'yxatiga ega.

Har bir fayl ob'ekti murojaat maskasiga ega (access mask-maskasiga (dostupa). Murojaat maskasi o'z ichiga standart (standart), maxsus (specific) va qarindoshlik murojaat xuquqlariga ega:
- standart murojaat xuquqi-hamma himoya qilingan ob'ektlar uchun umumiy amallarni aniqlaydi;

- maxsus murojaat xuquqi, faylni ob'ektlarga mos bo'lgan asosiy xuquqlarni ko'rsatadi;

- Qarindoshlik murojaat xuquqlari tizimi tomonidan foydalaniladi, ular standart va maxsus xuquqlar kombinatsiyasini aniqlaydi.

NTFS ruxsatlari Windows NT 4.0 va Windows 2000 /XP operatsion tizimlari uchun ruxsatlar har xil keltirilgan.

NTFS ning Windows NT 4.0 dagi ruxsatlari. NTFS ning fayl va kataloglarga murojaati Windows 4.0 dagi ruxsatlari individual, standart va maxsus bo'ladi.

Individual ruxsatlar. Foydalanuvchiga u yoki bu tipdagi murojaatni berish imkoniyati individual ruxsatlar deyiladi. Windows NT 4.0 da bunday ruxsatlar 6 ta: o'qish (read), yozish (write), bajarish (execute), olib tashlash (delete), ruxsatlarni o'zgartirish (Change Permissions) va egasini o'zgartirish (Take Ownership).

Standart ruxsatlar. har gal individual ruxsatlarning kombinatsiyasidan foydalanmaslik uchun, NTFS ning standart ruxsatlari kiritilgan.

- List - ko'rish

- Add - qo'shish

- Add & Read - o'qish va yozish

- Change - o'zgartirish

- Full Control - to'liq murojaat.

Maxsus ruxsatlar. Bu individual ruxsatlarning standart ruxsatlar bilan mos kelmaydigan kombinatsiyalari: R, W, X, D, P va O.

NTFS ruxsatlarini qo'llash. NTFS ruxsatlari foydalanuvchilar va uruhlar hisob yozuvlariga, xuddi umumiy Tarmoq resusrlariga murojaat ruxsatlari kabi beriladi. Foydalanuvchi ruxsatni bevosita yoki ruxsati bo'lgan bitta yoki bir nechta guruh a'zosi bo'lib olishi mumkin.

Windows 2000 va Windows XP OT larida individual, standart va maxsus ruxsatlar o'rniga, foydalanuvchi interfeysda 13 ta ruxsatlar mavjud.

NTFS fayl sistemasi 90-yillarning boshida FAT fayl sistemasi o'rnida Windows uchun asosiy fayl sistemasi sifatida yaratilgan. NTFS ning asosiy farqlanuvchi xususiyatlari quyidagilar:

1. Juda katta fayllar va juda katta disk hajmini qo'llab quvvatlash

2. Uzilishlar va dasturiy va apparat rad etilishlar keyin qayta ishlatilish xususiyati

3. Katta o'lchamdagi disklar uchun ham yuqori amallar tezligi

4. Fragmentatsiya darajasining pastligi katta disklar uchun ham

5. Egiluvchan struktura

6. Disk qurilmalarini rad etishga bardoshlilik

7. Uzun simvolli nomlarni qo'llab quvvatlash

8. Katalog va alohida fayllarga kiritishni boshqarish

FAT dan farqli ravishda NTFS ning barcha qismlari o'zida yoki fayl va fayl qismini qabul qiladi. NTFS ning asosiy strukturasini fayllarning asosiy jadvali hosil qiladi. Bugungi kunda qo'llanilayotgan ko'pchilik disklar uchun NTFS ning yozish hajmi 2 Kb ni tashkil qiladi. NTFS da barcha fayllar indesifikatsiyalanadi. Fayllarning bunday indesifikatsiyalash usuli oldingi fayl sistemalarida foydalanuvchi indesifikatsiyaga o'xshash

NTFS ning butun tomi klasterlar ketma - ketligidan tashkil topgan. Bu uni oldin ko'rib chiqilgan fayl sistemalaridan farqlaydi. NTFS da klasterlarning tartib raqami klasterning logik nomeri deb ataladi. NTFS fayli ham klasterlar ketma - ketligidan tashkil topgan. Bunda fayl ichidagi klaster tartib raqami klasterning virtual nomeri deb nomlanadi.

NTFS da klaster raqamlarini saqlash uchun razryadli ko'rsatkichdan foydalaniladi.

NTFS fayl strukturasi. NTFS tomida har bir fayl va katalog atribut jamlanmasidan tashkil topadi. Ta'kidlash kerakki fayl nomi va uning ma'lumotlari fayl atributlari sifatida qaraladi. Ya'ni NTFS traktida fayl atributidan tashqari boshqa har qanday komponentlar mavjud emas. NTFS fayli atributlari quyidagi maydonlardan tashkil topgan.

Atribut turi, atribut uzunligi, atribut qiymati va atribut nomi bo'lishi mumkin

Joylashgan usuliga bog'liq holda NTFS da fayllar unchalik katta bo'lmagan kattaliklarga ajratiladi. Ungacha katta bo'lmagan hajmga ega bo'lsa u holda u butunligicha NTFSning bitta yozuvi ichiga joylashtiriladi. Agar sig'masa bu haqidagi ma'lumot Data atributining bosh qismida namoyon bo'ladi.

10.4.4. ext2, ext3, ext4 va boshqa fayl tizimlari.

Linux operastion tizimi quyidagi fayl tizimlariga ega.

Ext – Linux operastion tizimning birinchi fayl tizimi xisoblanadi, dastlabki linux versiyalarida qo'llanilgan. Jurnallar bilan ishlay olmaydi, mana shu uning kamchiligi hisoblanadi.

Ext2 – 1993 yilda yaratilgan eski fayl tizimlaridan biri hisoblanadi. Ko'p yillar davomida bu fayl tizimi barcha distributivlarda asosiy fayl tizimi sifatida qo'llanilgan. Ext2 fayl tizimini ishlashi kesh xotira bilan bog'liq. Barcha disklarda bo'layotgan jarayonlarni kesh xotiraga tashlab, so'ng diskga tashlanadi. Shu sababli bu fayl tizimining tezligi juda baland, lekin ish jarayonida tok biror sababga ko'ra o'chib qolsa, kesh xotiradagi barcha ma'lumotlarni yo'qotib qo'yishimiz mumkin bo'ladi. Chunki qattiq diskga yozilmagan ma'lumotlar kesh xotiradan o'chib ketadi. Shu sababli bunday fayl tizimi bilan ishlayotganda uzluksiz tok manbaidan foydalanish zarur. Hozirgi kunda bu fayl tizimi o'zining muhimligini yo'qotgan.

Ext3 (Third Extended Filesystem)– ext2 fayl tizimining yangilangan ko'rinishi, lekin bu tizimda yangi texnologiya (jurnallar bilan ishlash) qo'llanilgan. Diskda sodir bo'lgan o'zgarishlarning barchasi "jurnallarga" yozib boriladi va ma'lumotlarga zarar yetkanda shu jurnallar orqali fayllarni tiklash mumkin bo'ladi. Jurnallar bilan ishlashning quyidagi rejimlari mavjud:

- **Ordered** – bu rejimda fayl tizimida sodir bo'lgan hizmat(shujebno'y) fayllarini o'zgarishlari yozib boriladi. Rejim odatiy tarzda ishlatiladi.

- **Journal** – buzilish sodir bo'lganda ma'lumotlarni minimal yo'qotishga erishmoqchi bo'lsangiz shu rejimdan foydalanishingiz mumkin. Bu rejimda hizmat fayllaridan tashqari foydalanuvchining ma'lumotlari xam jurnalga yozilishi mumkin. Shuning uchun bu rejim eng sekin ishlaydigan rejim hisoblanadi.

- **Writeback** – eng tez va eng foydasiz bo'lgan rejim hisoblanadi.

Jurnalning ishlash rejimini faqat Ext3 fayl tizimi uchun o'rnatish mumkin va u quyidagicha o'rnatiladi: `/etc/fstab` fayli o'zgartiriladi `/dev/sda5 / ext3 defaults, datajournal 1 1`

Ext3 fayl tizimida disk razdellarining maksimal qiymati 4 Tbayt, lekin linux yadrosining 2.6 versiyasida 16 Tbayt qilib belgilangan.

Faylning maksimal xajmi 1 Tbayt bo'lishi mumkin. Jurnallar bilan ishlash texnologiyasini o'chirib qo'yish xam mumkin.

Ext4 – Linuxning yangi ishlab chiqilgan fayl tizimlaridan biri hisoblanadi. Bu fayl tizimi Linuks yadrosining 2.6.28 versiyasida paydo bo'lgan. Ext3 fayl tizimi bilan solishtiradigan bo'lsak, Ext4 fayl tizimi ishlashi va ishonchiligi 2 barobar oshganligini ko'rishimiz mumkin. Bu tizimda disk razdelining maksimal qiymati 1024Pbayt (1Ebayt), fayl xajmining maksimal qiymati 2 Tbayt qilib belgilangan.

ReiserFS – Jurnallar bilan ishlay oladigan fayl tizimlaridan biri xisoblanadi. Bu fayl tizimi asosan kichik xajmdagi fayllar bilan ishlashda foydali. Bu fayl tizimining o'ziga xos tomoni shundan iboratki, bir necha kichik xajmdagi fayllarni bir blokda saqlash mumkin. Misol uchun, blok o'lchamingiz 4 Kbayt, siz bu blokga 1 Kbaytli 4 ta faylni joylashingiz mumkin. Shu tariqa siz diskdagi bo'sh shoylaringizni tejashingiz mumkin bo'ladi. Katta fayllar bilan bu fayl tizimi juda sekin ishlaydi va bu tizim uzilishlarga(tokning uzilib qolishi, kompyuterni o'chib qolishi) bardoshli emas, shu sababli ReiserFS fayl tizimini tez-tez defragmentatstiyadan o'tkazib turishingiz lozim. Agar siz ushbu fayl tizimidan foydalanmoqchi bo'lsangiz, albatta doimiy tok manbai (UPS) dan foydalaning.

XFS – Dastlab bu tizim **Silicon Graphics(SGI)** kompaniyasi tomonidan **Irix** operastion tizim uchun yaratilgan. Bu tizim ham jurnallar orqali ishlaydigan fayl tizimi xisoblanadi. Jurnallarga fayl tizimining umumiy metama'lumotlari yoziladi va buzilish sodir bo'lganda jurnaldagi ma'lumotlarning fayl tizimiga nusxasi olish orqali qayta tiklashni amalga oshirish mumkin bo'ladi. Jurnal hajmi fayl tizimini xosil qilishda o'rnatiladi, xajm 32 megabaytdan kam bo'lmasligi lozim. Bu fayl tizimi ext3, ReiserFS va JFS fayl tizimlariga nisbatan tezroq ishlaydi, lekin ext4 fayl tizimidan sekinroq. XFS fayl tizimiga katta hajmdagi bloklarni (64 Kbayt gacha) o'rnatish mumkin. Shuning uchun bu fayl tizimida video fayllarni qayta ishlash kabi grafikli uzellarda ishlatish tavsiya etiladi.

JFS – dastlab bu tizim **IBM** kompaniyasi tomonidan **AIX** operastion tizim uchun ishlab chiqilgan, keyinroq esa OS/2 uchun va nihoyat Linuks operastion tizim uchun ishlatilmoqda. Bu ham jurnall orqali ishlaydigan fayl tizimi. Jurnalning hajmi taxminan umumiy fa tizimining 40% tashkil etadi. JFS tizimi jurnal va ma'lumotlardan ibor

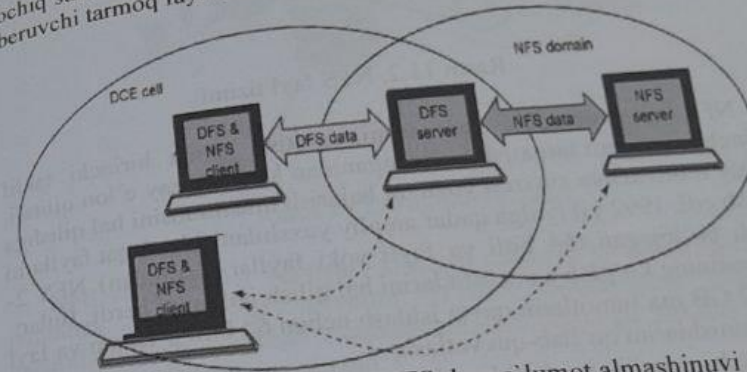
bo'lgan bir necha segmentlardan tashkil etilishi mumkin. Bu segmentlar agregatlar deyiladi va bu agregatlarni alohida montirovka qilish mumkin.

10.5. Nazorat savollari

1. Klaster nima?
2. Fayl tizimi nima?
3. Fayl tizimlaridan foydalanish nima berayapti?
4. Qanday fayl tizimlari ShK da ishlatiladi?
5. FAT fayl tizimi umumiy qurilish prinsiplari
6. FAT jadvali nimadan iborat?
7. HPFS tizimi ishlash prinsiplari?
8. NTFS fayl tizimi imkoniyatlari.
9. Standart, individual va maxsus ruxsatlar.
10. VFAT va FAT 32 fayl tizimlar
11. Fayl tizimi, ma'lumotlar ierarxiyasi.

11. VIRTUAL FAYL TIZIMLARI (VFS) VA TARMOQ FAYL TIZIMLARI (NFS)

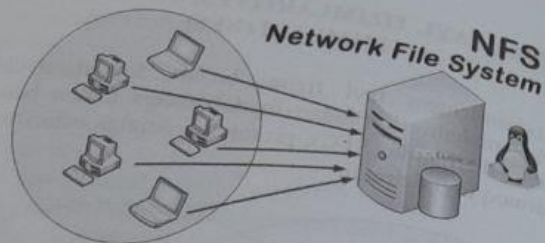
NFS taqsimlangan fayl tizimi bo'lib, foydalanuvchi mijoz kompyuterida mahalliy xotiradan foydalanishga imkon beradi. NFS ochiq standart sharhlarda (RFC), protokolni amalga oshirishga imkon beruvchi tarmoq fayl tizimidir.



Rasm 11.1. DFS va NFS da ma'lumot almashinuvi

NFS da 2-versiya dastlab faqat Ucer Datagram Protokol (UDP) protokoli bo'yicha ishlagan. U asosiy protokoldan tashqarida blokirovka qilinmagan server tomonini saqlash maqsadida ishlab chiqilgan.

Virtual fayl tizimining interfeysi oddiy protokolda aks ettirilgan modulli efirmi bajarishga imkon beradi. 1986 yil fevral holatiga ko'ra, tizim DOS va VMS kabi eysdan foydalangan holda ishlatiladigan operatsion tizimlar uchun yechimlar namoyish etildi. NFS v2 32bit cheklovlar tufayli faqat birinchi 2 Gb ni o'qishga ruxsat berdi.



Rasm 11.2. NFS fayl tizimi

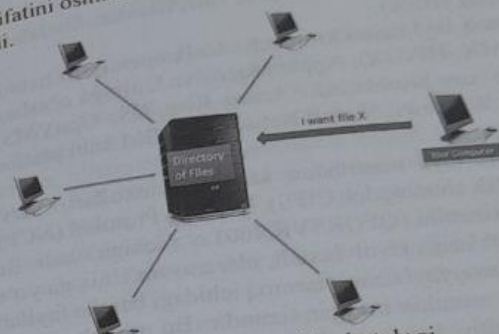
NFS ning uchinchi versiyasini yaratish uchun birinchi taklif ikkinchi versiyani tarqatish boshlanganidan ko'p o'tmay e'lon qilindi. Asosiy motivatsiya sinxron yozuvni bajarish muammosini hal qilishga urinish edi. 1992 yil iyulga qadar amaliy yaxshilanishlar faqat fayllarni yetarli bo'lmagan (64 bitli va fayli yoki fayllar hisobidan) NFS 2-versiyasining ko'plab kamchiliklarini hal qilish imkonini berdi. Bular:

- 2 GB ma'lumotlarni qayta ishlash uchun 64 bitli o'lcham va fayl almashinishlarini qo'llab-quvvatlash;
- mahsuldorlikni oshirish uchun serverda asinxron qayd qilishni qo'llab-quvvatlash;
- faylning qo'shimcha atributlari ularni qayta rasmiylashtirish zarurati oldini olish uchun ko'plab imkoniyatlar;
- ma'lumotlar va atributlar uchun ma'lumot va sifatlardan fayl nomlari bilan bir qatorda foydalanish;
- boshqa ko'plab yaxshilanishlar.

3-versiya kiritish jarayonida transport darajasidagi protokol sifatida TCP qo'llab-quvvatlashi osha boshladi. TCP-dan foydalanish UNS orqali NFS orqali foydalanilgan ma'lumotlar uzatish vositasi sifatida, o'rish va yozish uchun katta fayl o'lchamlarini uzatishga ruxsat berildi. Buning evaziga ishlab chiquvchilar Foydalanuvchi Datagram protokoli (UDP) tomonidan o'rnatilgan 8 Kb hajmdagi cheklovlar irasini yengishga muvaffaq bo'lishdi.

NFS v4. 4-versiya Enterra fayl tizimi (AFS) va Server xabarlarini kining ta'siri ostida ishlab chiqilgan, ish faoliyatini oshirishni o'z

ichiga oladi, sifatini oshiradi va belgilangan shartlarga muvofiq ravishda kiradi.



Rasm 11.2. Fayllar katalogi.

NFS versiyasi 4.1 versiyasi klasterli serverni joylashtirishni, shu jumladan bir nechta serverlar (PNFS kengaytmasi o'rtasida taqsimlanadigan fayllarga taqqoslanadigan) fayllarga parallel kirish imkoniyatini berishga qaratilgan.

Yangi fayl tizimi protokoli - NFS 4.2 (RFC 7862) - 2016 yil noyabr oyida rasmiy ravishda chiqarildi.

Standartning rivojlanishi bilan, u bilan ishlash uchun mos keladigan vositalar paydo bo'ldi. Shunday qilib, 2 va 3-versiyalar uchun veb-tafsilotlar tarmog'iga kirish protokoliga fayl tizimlariga ulanish va xavfsizlik parametrlari orqali ishlashni osonlashtirishga imkon beradi.

Uchinchi versiya guruhlarining turli protokollari NFS bilan ham bog'liq. Bularning eng mashhuri:

- Network Lock menejeri (NLM) bayt protokoli bilan qo'llab-quvvatlash (UNIX tizimini qo'llab-quvvatlash uchun qo'shilgan);
- NFS foydalanuvchilariga NFS serverlarida ma'lumotlarni saqlash kvotasini ko'rishiga imkon beradigan masofaviy kvota (RQOTAD);
- RDMA - Masofadan to'g'ridan-to'g'ri foydalanish (RDMA) orqali ishlatiladigan moslashish.

• NFS-GANESHA - bu foydalanuvchi kosmosida ishlaydigan va Ceffs-ni qo'llab-quvvatlaydigan, libcefflardan foydalanadigan FSAL (favqulodda yozuv).

Tarmoq fayl tizimi ko'pincha Unix operatsion tizimlari (masalan Solaris, AIX, HP-UX), Apple MacOS va Unix-ga o'xshash Operatsion tizimlarga xos hisoblanadi. Acorn Risp OS, OpnVMS, MSS-DOS, Microsoft Windows, Novell Netware va IBM kabi platformalar uchun mavjud.

Muqobil fayl masofadan kirish protokollari, Server xabarlar blokini (SMB, shuningdek CIFS), Netware Protokol (NCP) va OS / 400 Server fayl tizimini (QFORSVR.400) o'z ichiga oladi. Buning sababi NFSning talablariga javob beradi, ular asosan Unix ga yo'naltirilgan.

NFS (*tarmoq fayl tizimi*) tarmoq ichidagi barcha fayllarni/katalogni tarqatishingiz mumkin bo'lgan tizimdir. Bu mijozlarga masofadan ish mahalliy katalog sifatida ulash imkoniyatini beradi. Ya'ni ulangan fayl bilan bir xil dasturning o'zi hech qanday mijozga muvaffaqiyat bilan ishlashi mumkinligini anglatadi.

Afzalliklari:

- protsessorga yukni qisqartirish;

- tizimdagi oddiy katalog sifatida umumiy resurslarni aks ettirish;

Hozirda mavjud NFS V4.1 yangi PNFS imkoniyatni taqdim etadi. Almashishni amalga oshirishga imkon beradi. NFS 2 va 3 uchun kengaytma mavjud- WebNFS. Bu veb-brauzerlarga qo'shilishni va xavfsizlik devori orqali ishlash imkonini beradi.

11.1. Tashqi xotirani boshqarish

Tashqi xotirani boshqarish (EMM) plugin interfeysi CUDA Array interface CUDA qurilmalariga kiradigan turli xil ma'lumotlarni nashish imkoniyatini beradi. Biroq, har bir kutubxonaga o'z xotirasini shqalardan farqli ravishda boshqaradi. Masalan:

• Numba CUDA qurilmalarida CUDA drayveri API bilan o'zaro ishlashda bo'lib, cuMemAlloc va cuMemFree kabi funksiyalarni boshqarish uchun xotirani ajratadi, bu ko'p hollarda foydalanish uchun beradi.

• RAPIDS kutubxonalari (cuDF, cuML va boshqalar) qurilma xotirasini ajratish uchun RAPIDS Memory Manager (RMM) dan foydalanadi.

CuPy ikkala qurilma va mahkamlangan xotira uchun xotira fondini dasturini o'z ichiga oladi. Bir nechta CUDA-dan xabardor bo'lgan kutubxonalar birgalikda ishlatilganda, Numba xotirani boshqarish uchun boshqa ajratmalar va taqsimotlar uchun boshqa CUDA-xabardor bo'lgan kutubxonadan foydalanishga imkon berish orqali buni osonlashtiradi. Xotirani boshqarish uchun tashqi kutubxonadan foydalanishni osonlashtirish uchun EMM pluginidan foydalaniladi.

EMM plugini tashqi kutubxonaning bir qismi bo'lishi mumkin yoki alohida kutubxonaga amalga oshirilishi mumkin. Xotirani boshqarish uchun tashqi kutubxonadan foydalanishni osonlashtirish uchun tashqi xotirani boshqarish pluginidan foydalaniladi. Tashqi xotirani boshqarish plugini tashqi kutubxonaning bir qismi bo'lishi mumkin yoki alohida kutubxonaga amalga oshirilishi mumkin.

Tashqi xotirani boshqarish pluginidan foydalanilganda, Numba plugin orqali xotirani ajratadi. Hech qachon cuMemAlloc, cuMemFree va boshqalar kabi funksiyalarni to'g'ridan-to'g'ri chaqirmaydi.

Tashqi xotirani boshqarish pluginlari har doim qurilma xotirasini boshqarish uchun javobgarlikni o'z zimmlariga oladi. Shu bilan birga, CUDA-ni biladigan barcha kutubxonalar ham xost xotirasini boshqarishni qo'llab-quvvatlamaydi, shuning uchun Numba uchun xost xotirasini boshqarishni davom ettirish uchun qurilma xotirasini boshqarish tashqi xotirani boshqarishga berilganda ta'minlanadi.

Numba-ning ichki taqsimot xatti-harakatlari, muhim miqdor kutilmaguncha, taqsimotlarni kechiktirish orqali samaradorlikni oshirish uchun mo'ljallangan. Bundan tashqari, defer cleanup () kontekst menejeridan foydalanib, juda muhim bo'limlarda taqsimotlarning oldini olish mexanizmini taqdim etadi.

Tashqi xotirani boshqarish pluginidan foydalanilganda, ajratish strategiyasi Tashqi xotirani boshqarish tomonidan amalga oshiriladi va Numbaning ichki taqsimlash mexanizmi ishlatilmaydi. Tashqi xotirani boshqarish pluginini amalga oshirish mumkin:

- Numba ajratish xatti-harakatlariga o'xshash strategiya bilan;
- Plugin uchun ko'proq mos kelishi bilan.

Xotiradan tashqari Numba voqealar, oqimlar va modullarni taqsimlashni boshqaradi (modul bu @ cuda.jit -ted funksiyalaridan hosil bo'lgan kompilyatsiya qilingan ob'ekt). Hodisalar, oqimlar va modullarni boshqarish Tashqi xotirani boshqarish pluginidan foydalangan holda o'zgarmaydi.

Tashqi xotirani boshqarish pluginini BaseCUDAMemoryManager dan olish orqali amalga oshiriladi. Amalga oshirish uchun mulohazalarning qisqacha mazmuni quyidagicha:

- Numba har bir kontekst bo'yicha tashqi xotirani boshqarish Plugin sinfining bitta nusxasini yaratadi. Agar kerak bo'lsa, tashqi xotirani boshqarish Plugin ob'ektiga ega bo'lgan kontekstga self.context orqali kirish mumkin.

- Tashqi xotirani boshqarish plugini Numba ishlatadigan har qanday kod uchun shaffofdir - uning barcha usullari Numba tomonidan chaqiriladi va hech qachon Numba ishlatadigan kod bilan chaqirilishi shart emas.

- Memalloc, memhostalloc va mempin ajratish usullari qurilmani yoki xost xotirasini ajratish va / yoki pin qilish uchun asosiy kutubxonadan foydalanishi va Numba- ga qaytish uchun xotira ko'rsatgichining namunasini yaratishi kerak. Ushbu usullar har doim joriy CUDA konteksti tashqi xotirani boshqarish plugin nusxasiga tegishli kontekst bo'lganda chaqiriladi.

- Initialize tashqi xotirani boshqarish plugin ob'ekti kontekst uchun birinchi marta ishlatilishidan oldin Numba tomonidan chaqiriladi. Ushbu usul bir necha marta chaqirilishi mumkin va chaqirilganda avvalgi holat bekor qilinmasligi kerak.

- reset usuli kontekstdagi barcha ajratmalarni tozalash kerak bo'lganda chaqiriladi. Uni initialize dan oldin ham chaqirish mumkin, va tashqi xotirani boshqarish pluginini amalga oshirish bundan qatlanishi kerak.

11.2. Keshlash

Hisoblashni qayta ishlash sohasida kesh bu tezkor saqlash qatlami ib, unda kerakli ma'lumotlar to'plami odatda vaqtinchalik siyatga ega. Ushbu darajadagi ma'lumotlarga kirish asosiy saqlash a qaraganda ancha tezroq amalga oshadi. Keshlash yordamida

ilgari olingan yoki hisoblangan ma'lumotlarni samarali ravishda qayta ishlatish mumkin bo'ladi.

Keshdagi ma'lumotlar odatda tezkor kirishga ega bo'lgan qurilmada, masalan, RAM (tezkor kirish xotirasi) da saqlanadi va dasturiy ta'minot komponentlari bilan bo'lishi mumkin. Keshning asosiy vazifasi - ma'lumotlarni qidirish jarayonini tezlashtirish hisoblanadi. Bu sekinroq yotadigan saqlash darajasiga kirish zaruratini yo'q qiladi. Kesh xotirasining oz miqdori yuqori kirish tezligi bilan qoplanadi. Kesh odatda faqat kerakli ma'lumotlar to'plamini saqlaydi va ma'lumotlar bazalaridan farqli o'laroq vaqtincha, ma'lumotlar odatda to'liq va doimiy ravishda saqlanadi.

RAM va xotiradagi xizmatlar

RAM va xotiradagi xizmatlar so'rovlarni qayta ishlashning yuqori tezligini yoki IOPS ni ta'minlaganligi sababli, keshlash qidirish tezligini yaxshilaydi va katta hajmda ishlaganda xarajatlarni kamaytiradi. An'anaviy ma'lumotlar bazalari va qattiq disk uskunalarini bilan bir xil ko'lamga erishish uchun qo'shimcha manbalar talab qilinadi. Ushbu resurslardan foydalanish xarajatlarning ko'payishiga olib keladi.

Foydalanish sohalari. Kesh turli xil texnologik darajalarda, shu jumladan operatsion tizimlarda, tarmoq sathlarida, shu jumladan tarkibni yetkazib berish tarmoqlari (CDN) va DNS, Internet dasturlari va ma'lumotlar bazalarida qo'llaniladi. Keshlash kechikishni sezilarli darajada kamaytirishi va IOPS-ni ko'plab o'qish uchun intensiv dastur yuklari, masalan, savol-javob portallari, o'yin resurslari, ommaviy axborot tarqatish portallari va ijtimoiy tarmoq saytlari kabilarni yaxshilashi mumkin. Ma'lumotlar to'plamini qayta ishlash uchun intensiv ish yuklarini hisoblash masalan, tavsiya xizmatlari va yuqori mahsuldorlikni hisoblash modellashtirish ham xotiradagi ma'lumotlar qatlamidan kesh sifatida samarali foydalanishi mumkin. Ushbu dasturlar real vaqt rejimida yuzlab tugunlarni qamrab oladigan mashinalar klasterlari bo'yicha juda katta ma'lumotlar to'plamlariga kirishlari mumkin bo'ladi.

Tarqatilgan hisoblash muhitida ajratilgan keshlash qatlami tizimlar va dasturlarning keshdan mustaqil ravishda ishlashiga imkon beradi. Biroq, ularning hayotiy sikllari keshga ta'sir qilmaydi. Kesh

markazlashtiruvchi qatlam bo'lib xizmat qiladi, unga turli xil bog'liq bo'lmagan tizimlar kirishi mumkin. Uning o'ziga xos hayot sikli va me'moriy topologiyasi mavjud. Bu, ayniqsa, dastur tugunlari har ikki yo'nalishda ham dinamik ravishda masshtablashi mumkin bo'lgan tizimlar uchun juda muhimdir. Agar kesh dastur yoki bitta butunligini buzishi mumkin. Bundan tashqari, agar mahalliy keshlardan foydalanilsa, bu faqat ma'lumot iste'mol qiladigan mahalliy keshning foyda keltiradi. Taqsimlangan kesh muhitida ma'lumotlar bir nechta kesh-serverlarni qamrab olishi va markaziy joyda bo'lishi mumkin.

Keshlashning afzalliklari:

- Ilova ish faoliyatini yaxshilash;
- Ma'lumotlar bazasi xarajatlarini kamaytirish;
- Server tomonidagi yukni kamaytirish
- Prognozlangan ishlash;
- Ma'lumotlar bazalaridagi to'siqlarni yo'q qilish;
- O'qish qobiliyatini yaxshilash (IOPS).

11.3. Tranzaksiya asosida fayl tizimlar

Tranzaksiya NTFS fayl tizimlari va kataloglarni avtomatik tarzda yaratish, o'zgartirish, o'zgartirish va o'chirish imkonini beradi. Bitimlardan foydalanish operatsiyaning to'g'riligini ta'minlaydi; bir qator fayl operatsiyalarida (tranzaksiya sifatida amalga oshiriladi), agar barcha operatsiyalar muvaffaqiyatli bo'lsa, operatsiya amalga oshiriladi. Agar biron bir nosozlik bo'lsa, butun operatsiya orqaga qaytadi va ishlamay qoladi.

Tranzaksiya NTFS fayl tizimi Windows Vista-da kiritilgan Windows yadrosi komponenti bo'lgan Kernel Transaction Manager ustki qismida amalga oshiriladi, bu yadro ichidagi ob'ektlarning tranzaksiyasini ta'minlaydi. NTFS fayl tizimi bunday ma'lumotlarning blok sifatida qo'llab-quvvatlaydi va jurnalga past darajadagi operatsiyalari yozib oladi.

Tranzaksiya NTFS fayl tizimi imkoniyatlari:

- **Bitta faylda atomik operatsiyalari.** Bunga oddiy misol - faylni dasturdan saqlash; agar dastur yoki mashina faylni yozish paytida ishdan chiqsa, unda faylning faqat bir qismi yozilishi mumkin, natijada

fayl buzilgan bo'lishi mumkin. Agar faylning oldingi versiyasi ortiqcha yozilgan bo'lsa, bu juda muhim muammo bo'ladi, chunki ma'lumotlar yo'qolishi mumkin.

- **Bir nechta fayllarni o'z ichiga olgan atomik operatsiyalari.** Agar dastur bir vaqtning o'zida bir nechta faylni bir qator o'zgarishlar bilan yangilab turishi kerak bo'lsa, barcha kerakli fayl operatsiyalari bitta operatsiya sifatida bajarilishi mumkin, bu ishlamay qolganda bir-biriga mos kelmaydigan yangilanishlarni oldini oladi;

- **Bir nechta kompyuterlarni o'z ichiga olgan atomik operatsiyalari.** Bir xil operatsiyalarni bir nechta kompyuterlarda bajarish korporativ tarmoqdagi juda keng tarqalgan ma'muriy vazifadir; Tranzaksiya NTFS fayl tizimi o'zgarishning barcha mashinalarga muvaffaqiyatli tatbiq etilishini ta'minlash uchun tarqatilgan tranzaksiyalar koordinatori bilan birlashadi.

O'qish operatsiyalari bundan mustasno, Fayl tizimi fayllarini shifrlash bo'yicha operatsiyalar uchun tranzaksiya NTFS fayl tizimi dan foydalanish Windows Vista-da Service Pack 1 va Windows Server 2008-ga qadar qo'llab-quvvatlanmaydi.

11.4. Tarmoq fayl tizimlari va boshqa tizimlar

NFS. Tarqatilgan fayl tizimiga kirish foydalanuvchilarga tarmoqdagi boshqa kompyuterda joylashgan katalogni o'rnatishga imkon beradigan bir oz buyruqlarni talab qiladi. Quyosh mikroTizimlari bir necha yil oldin u deb nomlangan narsani tarqatishni boshlaganda, bu muammo bilan to'qnashdi *Masofaviy protsessorlar*. (RPCS) va NFS.

NFS matnli fayllarga kirish huquqi orqali vazifa bajaradi. Shuningdek NFSni tarqatish imkoni ham mavjud. Xavfsizlik tizimi protseduralari tarmoqni bajariladigan fayllarning zararli aralashuvidan himoya qilish uchun ishlatiladi.

NFS – RPC. NFS an'anaviy ravishda RPC dasturi sifatida aniqlanadi, bu NFS mijozi uchun TCP yoki boshqa ortiqcha yuklab bo'lmagan protokol tarmog'ini talab qiladi. Internet muhandislik vazifalarini tashkil etishni tashkil etish (ITF) RPC 1832 yildagi RPC uchun so'rovni e'lon qildi. NFS tomonidan amalga oshiriladigan boshqa bir kishi NFS tomonidan ishlatiladigan ma'lumotlar

formatlarini tavsifladi; U RFC 1831 yilda "Tashqi ma'lumotlar taqdimoti" (XDR) hujjat sifatida nashr etildi.

Boshqa RFCs shifrlash himoyasi va algoritmlar bilan NFS sessiyalarida autentifikatsiya qilish uchun ma'lumot almashish paytida ishlatiladigan algoritmlar bilan shug'ullanadi, ammo biz birinchi bo'lib asosiy mexanizmlar bilan bog'liq.

Ushbu RFC qaysi protokollarni NFS operatsiyasini taqdim etadi, ammo hozirda NFS qanday ishlashini tasvirlab bermaydi.

NFS 3 versiyasi. (NFSV3 versiyasi) shaklida nfs, u o'z holatini tejashga qodir (bu holat) va NFSV4 qutqaradi. Bugungi kunda bu asosiy iborasi kimnidir ajablantiradi, ammo NFS qurilgan Dunyo TCP/IP, chunki ularning ko'p qismi trafikni tahlil qilish va himoya tizimi dasturiy ta'minotini ishlab chiqaradigan kompaniyalarni saqlab qolmaydi.

NFSV3 bayonnomasi chekka kompyuterlardagi shaffof katalog maydonlarida bir nechta qo'shimcha protokollar bilan bog'liq, chunki ular ishlatiladigan fayl tizimlariga bog'liq emas. NFS har doim ham muvaffaqiyatga erisha olmadi. Yaxshi misol - Protokolni o'rnatish boshlang'ich fayl identifikatorini, tarmoqni blokirovka qilish bo'yicha boshqaruv protokoli faylni blokirovka qilish bilan shug'ullangan. Ikkala operatsiya ham NFSV3 protokoli ta'minlanmaganligi hozirgi holatga muhtoj. Shu sababli, xuddi shunday ma'lumotlar oqim mexanizmlarini aks ettirmagan protokoli darajasi o'rtasida murakkab o'zaro ta'sirlar bo'lib o'tdi. Bundan tashqari, agar siz Microsoft Windowsda fayl va katalogni yaratish u Unixdagi kabi mutlaqo boshqacha amalga oshirilsangiz, vaziyat yanada murakkablashadi.

NFSV3 protokoli yordamchi protokollarini taqdim etish uchun portlardan foydalanish uchun xizmat qiladi. Bu portlar, protokollarning barcha darajalari va barcha xavfsizlik masalalari bilan bog'liq jihatdan murakkab rasmga aylanadi. Bugungi kunda u bunday operatsiyaning modelidan voz kechildi va individual portlar orqali yordamchi protokollarni amalga oshirishni ilgari bajargan barcha operatsiyalar NFSV4 protokoli tomonidan bitta taniqli port orqali boshqariladi.

NFS 4 versiyasi. NFSV4 protokoli uning holatini saqlaydigan protokol. Bunday xatti-harakatlar bir nechta radikal o'zgarishlarga olib keldi. Yordamchi protokollarni chaqirish kerakligini, chunki foydalanuvchi darajasidagi jarayonlar bartaraf etildi. Buning o'rniga,

fayli ochilishidagi barcha operatsiyalar va juda ko'p RPC qo'ng'iroqlari liniyaning fayllari operatsiyalari sifatida amalga oshirildi.

Barcha NFS versiyalari har bir ishning har bir bo'limini RPC mijoz va server operatsiyalari kontseptsiyasida aniqladi. Har bir NFSV3 so'rovi juda ko'p sonli RPC qo'ng'iroqlari va chiqish uchun shunday deb ataladigan kontseptsiyani talab qildi. 4-versiya buni kompozit Fayl tizimida juda ko'p operatsiyalar mavjud bo'lgan operatsiyalar (aralashma). Albatta, to'g'ridan-to'g'ri effektiv RPC qo'ng'iroqlari va tarmoq orqali bir qancha miqdordagi ma'lumotlar, hatto har bir RPC qo'ng'irog'i, aksariyat darajada katta ishlarni amalga oshirgan bo'lsa ham, katta hajmdagi ishlarni amalga oshirishi mumkin. Taxminan hisoblab chiqilgan, NFSV3-dagi RPC-ga qo'ng'iroqlar murakkab RPC protseduralari bilan taqqoslaganda besh baravar ko'p mijoz-server o'zaro ta'sirini talab qildi.

NFS-dan foydalanish. Umuman olganda, NFS o'rnatish Samba o'rnatishga o'xshash jarayon. Server tomonida siz fayl tizimlari yoki eksport uchun kataloglarni aniqlaysiz yoki almashish; Mijoz bu kataloglarni moslashtiradi. Masofaviy NFS katalogini o'rnatgandan so'ng, ushbu katalog mahalliy fayllar tizimining boshqa kataloglari kabi mavjud bo'ladi. NFS serverdagi konfiguratsiyasi oddiy jarayon. Minimal darajada, eksport faylini yaratish yoki tahrirlash va NFS tizimlarini boshqarish kerak. hujjatlarida (bosh sahifasida) tasvirlangan. NFS server kirish va boshqa eksport faylidagi aniq formatga ega. Fayl tizimini tashkil qilish uchun va boshqa eksport faylini o'zgartirib va quyidagi umumiy formatda fayl tizimini (parametrlar bilan) tavsiflanadi:

Calcate1 (yoki Fayl tizimi) mijoz1 (vaript1, opsiyon2) mijoz2 (opsiyal variant2)

NFSni amalga oshirishingizni sozlash uchun bir nechta oddiy parametrlar mavjud. Bularga quyidagilar kiradi:

- **Xavfsiz:** Ushbu parametr (asl qiymati) NFS ulanishi uchun mavjud TCP/IP portidan foydalanadi, bu 1024 dan kam. Xavfsizlikni taqiqlash uni taqiqlaydi.
- **RW:** Ushbu parametr NFS mijozlariga kirish huquqini yozishga imkon beradi. Odatiy foydalanish faqat o'qish.

• **Async:** Ushbu parametr unumdorlikni oshirishi mumkin, ammo NFS serverini qayta ishga tushirish jarayonini qayta ishga tushirishda, shuningdek, NFS DEMONni to'xtatish protsedurasiz. Standart sozlash - sinxronlash.

• **No_wdelay:** Ushbu parametr yozuvni kechiktirishni o'chiradi. Agar Async rejimi o'rnatilsa, NFS ushbu parametrni e'tiborsiz qoldiradi.

• **Nohid:** Agar siz bitta katalogni boshqasiga o'rnatangiz, eski katalog odatda yashirin yoki bo'sh ko'rinadi. Bunday xatti-harakatni taqiqlash uchun yashirish parametriga ruxsat bering.

• **No_Subtree_check:** Ushbu parametrta e'tibor bermaslikni soxlamasligingiz mumkin bo'lgan ba'zi xavfsizlik tekshiruvlari uchun subchektorlarning nazoratini o'chirib qo'yadi. Standart parametr subdikatorlarni boshqarish uchun ruxsat berish.

NFS-da foydalanuvchilar namoyishida siz NFS hajmida ishlaydigan foydalanuvchi yoki foydalanuvchi bilan haqiqiy foydalanuvchi yoki foydalanuvchilarni aniqlashingiz mumkin. NFS foydalanuvchisi foydalanuvchi huquqlari yoki displeyni namoyish etadigan guruhga ega. NFS jildlari uchun bitta (umumiy) foydalanuvchi va guruhdan foydalanish ma'muriyatning muhim harakatlarisiz himoya va moslashuvchanlikni ta'minlaydi.

NFS-COUS-fayl tizimidagi fayllardan foydalanganda foydalanuvchiga kirish odatda siqilgan bo'ladi. Bu shuni anglatadiki, foydalanuvchi fayllarga ushbu fayllarga kirish huquqiga ega bo'lgan anonim foydalanuvchilar sifatida murojaat qiladi. Ushbu xatti-harakatlar, ayniqsa, ildiz foydalanuvchisi uchun juda muhimdir. Biroq, foydalanuvchiga uzoq tizimda ildiz yoki boshqa biron bir foydalanuvchi sifatida fayllarga murojaat qilish kerak bo'lgan holatlar mavjud.

Majburiy autentifikatsiya mexanizmlari. NFSV4 qat'iy autentifikatsiya qilish mexanizmlarini ishlab chiqardi. Kerberos navlari juda keng tarqalgan. Shuningdek, quyi infratuzilmaning ommaviy kalit mexanizmi (Lipkey) tomonidan qo'llab-quvvatlanishi kerak. NFSV3 kirish uchun standart Unix shifrlashdan boshqa narsani qo'llab-quvvatlamagan - bu katta tarmoqlarda asosiy himoya muammolarini keltirib chiqardi.

Kontrativ mexanizmlar va autentifikatsiya uslublari. NFSV4 autentifikatsiya mexanizmlari va uslublarini muzokaralar olib borish imkoniyatini berdi. NFSV3da ishlatiladigan shifrlash uslubini qo'lda aniqlashdan tashqari ko'proq narsani qilish imkonsiz edi. Shundan so'ng tizim ma'muri shifrlash va himoya protokollarini muvofiqlashtirishi kerak.

NFSV4 tarmoq fayl tizimi eng dolzarb va Linux tizimlariga almashtiradi. raqobatchi Windows va (hozir) Linuxdagi barcha navlarda mavjud bo'lganligi sababli, umumiy Internet-fayllar tizimi (SMB) deb hisoblash mumkin. Afs hech qachon katta tijorat ta'siriga ega emas; Ma'lumotlarning migratsiya va tarqalishiga yordam beradigan taqsimlangan fayl tizimlarining o'ziga xos ma'nosi ilova qilinadi. NFS linux versiyasidan foydalanishga tayyor, ammo Linux yadrosi yadrosining eng keng tarqalgan muvaffaqiyatsiz versiyalaridan biri Linux-ning eng keng tarqalgan muvaffaqiyatsiz versiyalaridan biri Linuxni kechiktirdi. Darhaqiqat, Linux oldin NFSV3-ni to'liq qo'llab-quvvatlay boshlaganidan oldin ko'p vaqt o'tdi.

NFSV4ning kelishi bilan, bu noqulaylik tezda yo'q qilindi va NFSV4 uchun to'liq qo'llab-quvvatlash nafaqat Solar, AIX va Freebsd-da amalga oshirildi.

NFS hozirda juda katta ustunlikka ega bo'lgan yetuk texnologiya hisoblanadi: u himoyalangan va qulay bo'lgan - bu xavfsizlikka kirish uchun bitta xavfsiz ro'yxatdan o'tish va uning imkoniyatlaridan foydalanish uchun qulay deb topildi, hatto fayllar va ilovalar joylashgan bo'lsa ham turli xil tizimlar. Foydalanuvchilarning tizim tuzilmasi ko'plab dasturlar turli xil operatsion tizimlarda va shuning uchun turli xil kompyuterlarda fayllardan fayllardan foydalanishini unutmang. NFS turli operatsion tizimlar bilan ishlashni osonlashtiradi fayl tizimlari semantika va samaradorlik xususiyatlari haqida tashvishlanmaydi.

Tarmoq fayllari tizimi - bu oddiy fayl tizimining yuqori qismida bu oddiy fayl tizimining yuqori qismida tarmoqni tarmoq orqali kirishga imkon beradi. Garchi NFS birinchi tarmoq tizimi emas bo'lsa-da, bugungiXTda eng funktsional va talab qilingan tarmoq fayl tizimi ishlab chiqilgan. NFS foydalanuvchilarning ko'pligi uchun umu

Fayl tizimiga kirish va saqlash uchun zarur bo'lgan disk maydonini minimallashtirish uchun ma'lumotni markazlashtirishni ta'minlash imkonini beradi.

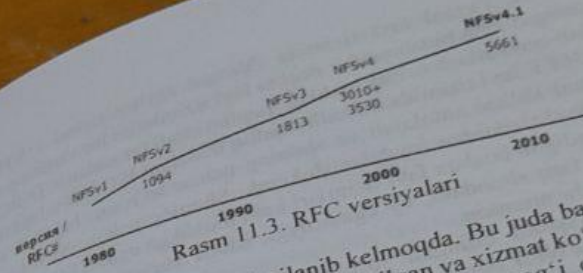
Ushbu maqola NFS tarixini qisqacha ko'rib chiqishdan boshlanadi va keyin NFS arxitekturasini o'rganishga va uni yanada rivojlantirishdan boshlanadi.

NFS ning qisqacha tarixi. Birinchi tarmoq fayl tizimi (faylga kirish tinglovchilari - faylga kirish ishlovchisi) deb nomlangan va 1976 yilda Dekabr tomonidan (raqamli uskunalar korporatsiyasi) tomonidan ishlab chiqilgan. Bu DAP Protokolini (ma'lumotlarga kirish protokoli - ma'lumotlar ulanish protokoli) amalga oshirishi va deputat protokoli - paketining bir qismi edi. TCP/IP holatida bo'lgani kabi, uning protokoli protokollari, shu jumladan DAP Protokolini nashr etdi.

NFS IP protokoli tepasida qurilgan birinchi zamonaviy tarmoq tizimi edi. Uning prototipi 80-yillarning boshlarida virtual mikro-sizimlarida ishlab chiqarilgan tajriba fayl tizimini quyosh mumkin. Ushbu echimning mashhurligini hisobga olgan holda, NFS protokoli RFC spetsifikatsiyasi sifatida va keyinchalik NFSV2da paydo bo'lgan. Nfs boshqa mijozlar va serverlar bilan o'zaro aloqada bo'lish imkoniyatlari tufayli tezda tasdiqlangan.

Keyinchalik, standart NFSV3 versiyasiga NFSV3 versiyasiga yangilandi. Protokolning ushbu versiyasi avvalgisiga qaraganda ko'proq tarqalgan va yirik fayllarni (2 Gb) va TCP transport protokoli sifatida qo'llab-quvvatlaydi. NFSV3 global (WAN) tarmoqlari uchun fayl tizimlarini rivojlantirish yo'nalishini so'radi. 2000 yilda RFC 3010 spetsifikatsiyasi doirasida (RFC 3530), NFS korporativ muhitga o'tkazildi. Quyoshni himoya qilingan NFSV4 bilan himoyalangan (oldingi NFS versiyalari shtatni saqlab qolishni qo'llab-quvvatlamadi, I.E. toifadagi ittifoqsiz). Hozirgi vaqtda NFS-ning so'nggi versiyasi 4.1-versiya bo'lib, RFC 5661-da aniqlanadi, unda protokolda kengayish orqali PNFS. Taqsimlangan serverlar uchun parallel kirish uchun qo'llab-quvvatlash qo'shildi.

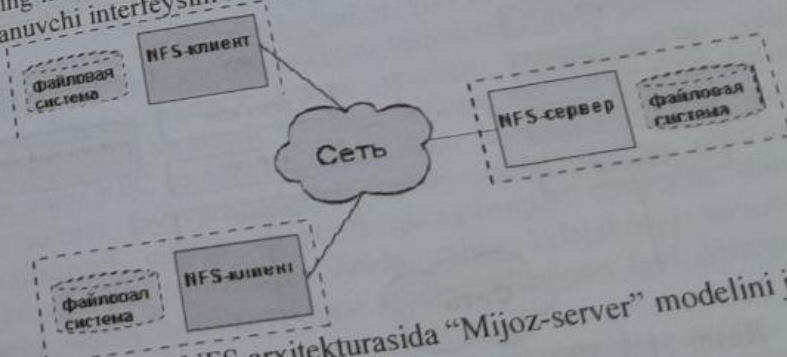
NFS rivojlanish tarixi, shu jumladan maxsus RFC, uning versiyalarini tavsiflovchi, 1-rasmda keltirilgan.



Rasm 11.3. RFC versiyalari

Deyarli 30 yil davomida rivojlanib kelmoqda. Bu juda barqaror va tarmoqli fayllar tizimi juda barqarorlashtirilgan va xizmat ko'rsatish va xizmat ko'rsatish sifati bilan o'tkaziladi. NFS tarmog'i doirasidagi ma'lumotlar almashish tezligini oshirish va kechikishlarni kamaytirish sharoitida NFS tarmog'i doirasida tarmoq ichidagi fayl tizimini amalga oshirishning mashhur usuli bo'lib qolmoqda. Mahalliy tarmoqlarda ham virtualizatsiya virtual mashinalar uchun qo'shimcha harakatchanlikni ta'minlash uchun tarmoqdagi ma'lumotlarni saqlashni qo'llab-quvvatlaydi. NFS shuningdek, virtual infratuzilmani optimallashtirishga qaratilgan hisoblash muhitini tashkil etish uchun eng so'nggi modellarni qo'llab-quvvatlaydi.

NFS arxitekturasini NFS standart mijoz-server arxitektura modelini (2-rasmda ko'rsatilganidek) ishlatadi. Server mijozlar almashinuvi va omborxonalar tizimini joriy qilish uchun javobgardir. Mijoz mijozning mahalliy fayl maydoniga o'rnatilgan umumiy fayl tizimiga foydalanuvchi interfeysini amalga oshiradi.

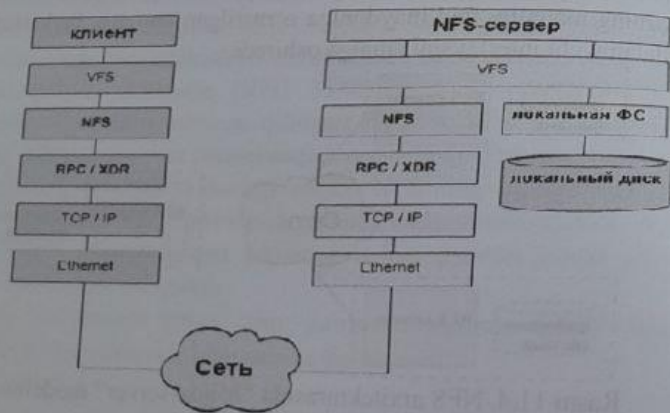


Rasm 11.4. NFS arxitekturasida "Mijoz-server" modelini joriy etish

Linux Virtual fayl tizimida (Virtual fayllar tizimi - VFS) bir vaqtning o'zida bitta xost bir nechta fayl tizimlarini bir vaqtning o'zida qo'llab-quvvatlash uchun mablag taqdim etadi (masalan, CD-ROM va ERD36FT fayl tizimida mahalliy qattiq diskdagi ISO 9660 fayl tizimi).

Virtual kalitni aniqlaydi va shuning uchun so'rovni bajarish uchun qaysi fayl tizimidan foydalanish kerak. Shuning uchun, NFS Linux va yagona farq shundaki, uy egasi uyga ishlov berish o'rtasidagi yuborilishi mumkin.

VFS qabul qilingan so'rov NFS-ni anglatadi va uni yadroda joylashgan NFS boshqaruvchisiga o'tkazadi. NFS ishlovchisi kirish / chiqish so'rovini ko'rib chiqadi va uni NFS protsedurasiga ko'rsatadi (ochiq, kirish, yaratish, yaratish, olib tashlash, olib tashlash va boshqalar). RFCning alohida spetsifikatsiyasida ko'rsatilgan ushbu protseduralar NFS protokolining xatti-harakatlarini aniqlang. Kerakli protsedura so'rovga qarab tanlangan va RPC texnologiyasi yordamida amalga oshiriladi (masofadan boshqarish pulti). Siz ismini tushunganingizdek, RPC turli xil tizimlar o'rtasida protseduralarni taklif qilishga imkon beradi. RPC xizmati nfs so'rovini o'z dalillari bilan bog'laydi va natijani tegishli masofaviy xostga yuboradi, so'ngra uni so'rovni qayta tashabbuskorga qaytarish va qayta ishlash uchun javob qaytaradi.



Rasm 11.5. VFS va NSF bog'lanish jarayoni diagrammasi

NFSv4-da keng kutish vaqti bilan tarmoqlar uchun maxsus kompozitsion protsedura taklif etiladi (murakkab protsedura.). Ushbu protsedura tarmoq so'rov yuborish narxini minimallashtirish uchun bitta so'rov ichida bir nechta RPC qo'ng'iroqlarini joylashtirishga imkon beradi. Shuningdek, ushbu protsedurada javob olish uchun qo'ng'iroqlarni qaytarish funksiyalari mexanizmi amalga oshirildi.

NFS protokoli. Mijoz NFS bilan ishlay boshlaganda, birinchi harakat - masofadan ish tizimining mahalliy fayl tizimiga o'rnatiladigan maydonchadir. Ushbu protsedura tarmoq so'rov yuborish narxini yo'naltirilganligini o'rnatadi (Linux tizim funktsiyalaridan biri). Keyin, RPC qo'ng'iroq'idan foydalanib, pultdagi get_port funktsiyasi montetring uchun so'rov yuboradi. Ushbu server so'rovi CUMC COPC monteting uchun so'rov yuboradi. Ushbu server so'rovi funktsiyasi ishlanadi. Maxsus tekshirish mijoz tomonidan talab qilingan fayl tizimi ushbu serverda mavjud bo'lgan tizimlar ro'yxatida mavjudligi bilan aniqlanadi. Agar so'ralgan tizim mavjud bo'lsa va mijoz unga kirish huquqiga ega bo'lsa, unda RMC protsedurasida protsessorning tavsifi ko'rsatiladi. Mijoz mahalliy va masofaviy o'rnatish haqida ma'lumotni tejaydi va I/O so'rovlarini bajarish mumkin. Protokol tatbiqi xavfsizlik nuqtai nazaridan benuqson emas, shuning uchun NFSv4da ichki RPCning ichki qo'ng'iroqlaridan foydalaniladi, bu esa ulanish punktlarini boshqarishi mumkin.

Fayl masofaviy fayl tizimida aniqlangandan so'ng, mijoz protsedura yo'q, buning o'rniga mijoz ko'rsatilgan fayl va katalog o'rnatilgan fayl tizimida mavjudligini tekshiradi. Mijoz GetTR RPC ma'lumotidan boshlanadi, bunga javoban katalog atribut yoki indikatorga javob bermaydi. Keyingi, faylning mavjudligini tekshirish uchun mijoz RPC so'rovini qidiradi. Agar fayl mavjud bo'lsa, getatt RPC so'rovi fayl atributlarini topish uchun amalga oshiriladi. Muvaffaqiyatli qidiruv va getatt qo'ng'iroqlari natijasida olingan ma'lumotlardan foydalangan holda, mijoz kelajakdagi so'rovlarni bajarish uchun foydalanuvchiga taqdim etiladigan fayl destriktorini yaratadi.

hisob-kitob, status, ofset va baytdan iborat. Mijoz hozirgi vaqtda operatsiya amalga oshirilishi mumkinligini aniqlash uchun status(holat)dan foydalanadi. Ofset (ofset.) Qaysi pozitsiyani o'qishni boshlashingiz va bayt peshtaxtasi (calculate) Qancha baytni ko'rib chiqish kerakligini aniqlaydi. RPC qo'ng'irog'i natijasida o'qish serveri har doim ham talab qilinganidek, baytlarni qaytarmaydi, ammo qaytarilgan ma'lumotlar har doim mijozga qancha bayt yuborilganini ko'rsatadi.

NFS-da innovatsiyalar. NFS - 4 va 4.1-ning so'nggi ikki versiyasi katta qiziqish uyg'otadi, bunda NFS texnologiyasi evolyutsiyasining eng muhim jihatlari o'rganilishi mumkin.

NFSV4 paydo bo'lishidan oldin, ularni boshqarish, blokirovka qilish va boshqa narsalar kabi fayllarni boshqarish vazifalari bajarilishi Maxsus qo'shimcha protokollar mavjud edi. NFSV4-da fayllarni boshqarish jarayoni bitta protokolga soddalashtirilgan; Bundan tashqari, UDP ushbu versiyasidan boshlab transport protokoli sifatida foydalanilmaydi. NFSV4-ni boshqa operatsion tizimlarga kiritish uchun NFSning "tabiiy" usulni o'z ichiga olgan, "Windows" semantikaga kiradi.

NFSV4.1da ishlash uchun katta miqyosli tushuncha joriy etildi. **parallel NFS (PNFS)**. Majburiyatning katta darajasini ta'minlash uchun NFSV4.1da ma'lumotlar va mebitatura amalga oshiriladi (**markalash**), tizim tarmoqlarga bilan taqsimlangan, chunki u klaster fayl tizimlarida amalga oshiriladi. Ko'rsatilganidek, PNFS uchta komponentga ekotizimni baham ko'radi: mijoz, server va xotira. Shu bilan birga ikkita kanal mavjud: birinchisi ma'lumotlar uzatish uchun, ikkinchisi boshqarish buyruqlarini uzatish uchun. PNFS ma'lumotlar larning metratasini tavsiflovchi usullardan ikki kanal arxitekturasini ta'minlaydi.

11.5. Nazorat savollari

1. Tashqi xotira bilan ishlash kutubxonalarini qaysilar?
2. NFS serverini sozlash parametrlari haqida tushuntirib bering?
3. NFS komponentlari haqida ma'lumot bering?
4. Ubuntu-da NFS serverini sozlashda ishlatiladigan buyruqlar?
5. NFS protokoli imkoniyatlari?

12. MA'LUMOTLARNI KIRITISH CHIQRISH TIZIMLARI

Operatsion tizim jarayonlar, manzillar fazolari va fayllardan tashqari kompyuterga ulangan barcha kiritish-chiqarish tizimlarini xam boshqaradi. OT qurilmalarga buyruq berib, uzulishlarni ushlashi va xatoliklarni qayta ishlashi zarur. Shuningdek, u qurilmalar va butun tizim o'rtasida ishlatishda sodda va oson interfeysni taqdim qilishi kerak. Interfeys imkon darajasida barcha qurilmalar uchun (qurilma turidan qat'iy nazar) bir xil bo'lishi lozim. Kiritish-chiqarish tizimi butun OTning ma'lum qismini tashkil qiladi[2].

12.1. Kompyuter tizimlarida kiritish-chiqarishni tashkil etish

Kiritish-chiqarish qurilmalarini shartli ravishda 2 kategoriyaga ajratish mumkin: blokli va simvulli qurilmalar, Blokli qurilmalarga axborotni belgilangan uzunlikdagi bloklarda saqlaydigan qurilmalar kiradi. Bloklar o'lchami odatda 512 baytdan 65536 baytgacha bo'ladi. Barcha ma'lumotlar paketlar ko'rinishida uzatilib bir yoki bir nechta (ketma-ket) bloklardan tashkil topadi. Eng keng tarqalgan blokli qurilmalar – qattiq disklar, Blu-ray disklar va USB flash-xotiralalar.

Kiritish-chiqarish qurilmalarining yana bir turi – simvulli qurilmalar. Ular xech qanday blokli strukturaga tegishli bo'lmagan simvulli ma'lumotlarni uzatadi va qabul qiladi. Bunday qurilmalarga printerlar, tarmoq interfeyslari, sichqonchalar va disk qurilmalarga o'xshamaydigan boshqa qurilmalar.

Ushbu tasniflash sxemasi mukammal emas. Ba'zi qurilmalar xechqaysisiga taaluqli emas. Masalan, soat na blokli na simvulli manzillashga taaluqlidir. Ularni asosiy qiladigan ishi – aniq vaqt oralig'ida uzulishlarni chaqiradi. Ekranlar va sensorli ekranlar xam shunday printsipga ega.

Kiritish-chiqarish qurilmalari tezliklarning katta oralig'ini qamrab oladi. Natijada ma'lumotlarni tez uzatish samaradorligini ta'minlashda dasturiy ta'minotga cheklovlar uchraydi. 1-jadvalda eng ko'p tarqalgan qurilmalarning ma'lumotlarni uzatish tezliklari keltirilgan.

1-jadval. Eng ko'p tarqalgan qurilmalarning ma'lumotlarni uzatish tezliklari

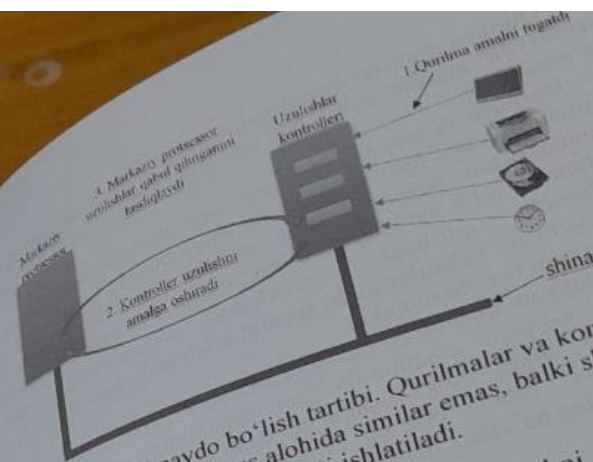
Qurilma	Ma'lumot almashish tezligi
Klaviatura	10 bayt/s
Sichqoncha	100 bayt/s
Raqamli kamera	3,5 Mbayt/s
802.11n standartli simsiz tarmoq	37,5 Mbayt/s
USB 2.0	60 Mbayt/s
SATA 3 diski	600 Mbayt/s
USB 3.0	625 Mbayt/s
SCSI Ultra 5 diski	640 Mbayt/s
PCI 3.0 shinaning 1 ta yo'lakchasi	985 Mbayt/s
Thunderbolt 2 shinasi	2,5 Gbayt/s
SONET OC-768 tarmog'i	5 Gbayt/s

12.2. Kontrollerlar

Kiritish-chiqarish qurilmalari ko'p hollarda mexanik va elektron qurilmalardan tashkil topadi. Elektron komponent qurilma kontrolleri yoki adapter deyiladi, Shaxsiy kompyuterlarda u tizimli platadagi mikrosxema yoki kengaytiruvchi (PCIe) slotlarga o'rnatiladigan platalar ko'rinishida bo'ladi. Mexanik komponenti sifatida qurilmani o'zi qatnashadi. Kontroller platasida odatda qurilmaga olib boradigan kabel ulanishi mumkin bo'lgan razyom bo'ladi. Ko'pgina kontrollerlar 2 ta, 4 ta xattoki 8 ta bir xil qurilmani boshqarish imkoniga ega.

Kontroller vazifasi ketma-ket bitlar oqimini baytlar blokiga aylantirishdan va zarur paytda xatolarni tog'irlashdan iborat. Baytlar bloki controller tarkibidagi buferda bitlar bo'yicha dastlab yig'iladi. Blokning nazorat summasi tekshirilgandan va unda xatolik yo'qligi e'lon qilingandan so'ng tezkor xotiraga nusxasi olinishi mumkin.

Shaxsiy kompyuterlar tizimida 12.1-rasmda tasvirlanganidek uzulishlar strukturasi mavjud. Apparat darajasida uzulishlar quyidagi tartibda ishlaydi.



12.1-rasm. Uzulishni paydo bo'lish tartibi. Qurilmalar va kontroller orasidagi aloqa uchun maxsus ajratilgan alohida similar emas, balki shinada uzulishlar chizig'i ishlatiladi.

Kiritish-chiqarish qurilmasi unga topshirilgan ishni tugatganda uzulishni amalga oshiradi (OT tomonidan ruxsat etilgan holda). Bu amal signalni maxsus ajratilgan shinaga uzatish yo'li bilan amalga oshiriladi. Tizimli platada joylashgan uzulishlar kontrolleri mikrosxemasi bu signalni aniqlabkeyingi harakatlar haqidagi yechimni qabul qiladi.

12.3. Drayverlar

Oddiy kiritish-chiqarish dasturiy ta'minoti to'rt darajali ko'rinishda tashkil qilingan (12.2.-rasm). Xar bir darajaning aniq ishchi funksiyasi va unga bog'langan darajalar o'rtasida o'zaro aniq interfeys mavjud.

12.2.-rasm. Kiritish-chiqarish dasturiy ta'minotining darajalari

Foydalanuvchi darajasi kiritish-chiqarish dasturiy ta'minoti
OTning qurilmaga bog'liq bo'lmagan dasturiy ta'minoti
Qurilmalar drayverlari
Uzulishlar kontrolleri
Apparatura

Qurilma drayveri operatsion tizim va boshqa dasturiy ta'minotni qandaydir apparat bilan bog'lanishni bildiradigan kichik dasturiy ta'minot.

Dasturiy ta'minot va apparat turli shaxslar yoki kompaniyalar tomonidan yaratilgan va ikkita butunlay boshqa tillarda gapirishgan, shuning uchun tarjimon ular bilan muloqot qilishga imkon beradi. Boshqacha qilib aytadigan bo'lsak, dasturiy ta'minot dasturi qanday qo'shimcha qurilmalar kerakligini tushuntirishga, qurilmani haydovchisini xabardor qilib, apparat bilan ishlashga imkon beradigan ma'lumotni berishi mumkin.

Muayyan apparat qurilmasiga xavfsiz holatga keltirilishi mumkin bo'lgan ko'plab muammolar asl apparatning o'zi bilan bog'liq muammo emas, balki ushbu apparat uchun o'rnatilgan qurilma drayverlari bilan bog'liq muammolar. Yuqorida bog'langan resurslarning ba'zilar sizning barchangizni tushunishga yordam berishi kerak.

Device Drivers. Bu kunlarda kamroq tarqalgan bo'lsa-da, ba'zi dasturiy ta'minot ba'zi bir turdagi apparat vositalari bilan bevosita aloqa qilish imkoniyatiga ega. Bu, odatda, dasturiy ta'minot apparatga juda oddiy buyruqlar yuborayotganda yoki ikkovini bir xil kompaniya tomonidan ishlab chiqilgan bo'lsa ham, lekin bu shuningdek, o'rnatilgan drayver holatiga o'xshash bo'lishi mumkin.

Ba'zi qurilma drayverlari bevosita qurilma bilan bog'lanishadi, ammo boshqalari bir-biri bilan qatlamlanadi. Bunday hollarda, dastur boshqa dastur bilan bog'lanishidan avval va dasturning aslida apparat bilan bevosita aloqani amalga oshirguniga qadar bir dastur bilan bog'lanadi.

Windows WindowsYD fayllarini yuklanishi mumkin bo'lgan qurilma drayverlari sifatida ishlatadi, ya'ni ularni har doim xotirada saqlamaydilar, shuning uchun kerakli darajada yuklanishi mumkin

WHQL Microsoft tomonidan muayyan qurilma drayverini Windowsning muayyan versiyasi bilan ishlayotganligini isbotlashga yordam beruvchi sinov jarayonidir.

Drayverning yana bir shakli - grafik qurilmalar bilan ishlaydigan virtual qurilma drayveri. Ular muntazam drayverlarga o'xshash ishlaydi, ammo mehmonlarning operatsion tizimini bevosita apparatdan foydalanishga yo'l qo'ymaslik uchun virtual drayverlar

haqiqiy apparat sifatida maskeleniyor, shunday qilib mehmonlar operatsion tizim va uning drayverlari virtual bo'lmagan operatsion tizimlarga o'xshash qo'shimcha qurilmalarga kirishadi.

12.4. Nazorat savollari

1. Axborotni kiritish chiqarish deganda nima tushuniladi?
2. Kiritish-chiqarish qurilmalari nima?
3. Qurilmalar kontrollerlari nima?
4. Uzulishlar nima?
5. Kiritish-chiqarish dasturiy ta'minoti darajalari qanday turlarga bo'linadi?

13. OPERATSION TIZIM VA TARMOQLARNING XAVFSIZLIGI

13.1. Xavsizlik konsepsiyasi

Ro'y berishi mumkin bo'lgan tahdidlarni bilish, va bu tahdidlar ta'sir qilish mumkin bo'lgan himoya joylarini bilish, xavfsizlikning eng tejamli vositalarini aniqlash uchun zarurdir.

Xavfsiz tizim, maxfiylik, murojaatlilik va yaxlitlik xususiyatlariga ega bo'lishi kerak. Har qayday maxfiylik, murojaatlilik va yaxlitlikni buzishga bo'lgan xarakteristik xarakteristik xarakteristik xarakteristik taxdid xujum deb ataladi.

Maxfiy (Confidentiality) tizim, maxfiy ma'lumotlarga faqat murojaat ruxsat berilgan foydalanuvchigina murojaat qila olishiga ishonchni ta'minlaydi. Bunday foydalanuvchilar mualliflashtirilgan deb ataladi.

Murojaatlilik (dostupnost-availability) deganda, mualliflashtirilgan (avtorizovanno'y) foydalanuvchilarga zarur ma'lumot doimo ochiqligi kafolatini tushunamiz. Va nihoyat tizim yaxlitligi (integrity) mualliflashtirilgan foydalanuvchi ma'lumotlarni hech qanaqasiga o'zgartira olmasligini nazarda tutadi.

Ma'lumotlar himoyasi, foydalanuvchi xatosi va qurilmalarning buzilish va x.k. kabi tasodifiy taxdidlardan farqlanuvchi va OT foydalanuvchilariga zarar etkazishni maqsad qilib olgan uyushtirilgan tahdidlar bilan kurashishga yo'naltirilgan.

Uyushtirilgan tahdidlar aktiv va passiv tahdidlarga bo'linadi. Passiv tahdid, ma'lumotlarga, tizim holatini o'zgartirmaydigan xuquqsiz murojaatdir, aktiv tahdid-tizimni xuquqsiz bo'lmagan holda o'zgartirishdir. Passiv tahdidni aniqlash qiyin, chunki ular ta'siri natijasida ma'lumotlar o'zgarmaydi. Passiv xujumdan ximoya, ularning oldini olishga asoslanadi.

Taxdidlarni bir nechta tiplarini ajratish mumkin. Eng ko'p tarqalgan tahdid- tizimga xuquqi bor (legal) foydalanuvchi sifatida kirishga harakat qilish, m-n, parolni topishga xarakteristik. Eng murakkab variant-tizimga, login so'zini ekranga chiqaruvchi dasturni kiritish. Ko'pgina legal xuquqi bor foydalanuvchilar bu holda tizimga kirishga xarakteristik qiladilar va ularning bu xarakteristiklari pratiklashtirilishi mumkin.

Bunday tashqaridan —zarasizl ko'ringan va kerak bo'lmagan funksiyalarni bajaruvchi dasturlar —Trojanskie konil deb ataladi. Ba'zida, bunday ortiqcha parolni tekshirish dasturlarini yo'qotish uchun, del, break, cancel va x.k. klavishalarni bir necha marotaba bosish kifoyadir.

Bunday xujumlardan himoyalani uchun OT foydalanuvchini —autentifikatsiyal qiluvchi jarayonni ishga tushiradi. Boshqa turdagi taxdidlar, legal foydalanuvchilarning mumkin bo'lmagan xarakteristiklari bilan bog'liqdir, ular m-n,oldinroq foydalanilgan ma'lumotlarni saqlab qolgan disk, hotira sahifalarini o'qishga xarakteristikdir. Himoya bu xolda mualliflashtirish (avtorizatsiya) ishonchli tizimiga asoslanadi. Bu kategoriyaga xizmat ko'rsatishni rad etish ham kiradi. Bu holda server so'rovnomalar bilan qo'shilib ketsa, ko'pgina mualliflashtirilgan foydalanuvchilar murojaatiga imkon bo'lmay qoldi.

Va nihoyat tizim ishi virus-dasturlar yoki —chervl- dasturlar yordamida buzilishi mumkin, ular maxsus ravishda kompyuter resurslaridan noto'g'ri foydalanish yoki zarar etkazishga mo'ljallangan bo'ladi. Bunday tahdidlarning umumiy nomi-zararli dasturlar (malicious software) dir.

Odatda ular, o'zicha tarqaladilar, kompyuterdan kompyuterchi zararlangan fayllar orqali yoki disketa va elektron pochta orqali o'tadilar. Bunday dasturlar bilan kurashishning samarali usuli — kompyuter gigienasiga rioya qilishdir. Ko'pfoydalanuvchili tizimlar, shaxsiy kompyuterlarga nisbatan bunday viruslardan kam zararlanadilar, chunki ularda himoyaning tizimli vositalari mavjuddir. Mana shular, axborot tizimlariga eng ko'p zarar keltiradigan dasturlardir.

13.2. Tarmoqli va tizimli taxdidlar(hujumlar)

Axborot xavfsizligi shunchalik katta ahamiyat kasb etdiki, qator mamlakatlarda axborot xavfsizligi muammosiga asosiy yondashishlar tartiblashtirilgan. Asos (osnovopolagayuhie) xujjatlar ishlab chiqildi.

Natijada axborot tizimlarini ishonchlilik darajasi bo'yici ranjirovka qilish (ajratish) mumkin bo'ldi.

Hammadan ham, AQSH mudofaa Vazirligining muqova rangiga qarab topilgan —oranjevayal kitobi taniqlidir (DoD 1993). Bu xujjatda xavfsizlikning 4 ta darajasi — D, S, V va A belgilangan. D darajada ga o'tish borasida tizim ishonchligiga qattiq talab qo'yib boriladi. A va V darajalar bir nechta sinflarga bo'linadi (S1, S2, V1, V2, V3). Tizimga sertifikatni berish jarayoni natijasida biror bir sinfga mansubligini belgilash talablarga javob berishi kerak.

Misol uchun S2 sinfi talablarini ko'rib chiqamiz. Bu talablarga Windows NT va UNIX ning ba'zi versiyalari javob beradi.

- Har bir foydalanuvchi tizimga kirishi uchun yagona nom va parol bilan ta'minlanishi kerak. Kompyuterga faqat autentifikatsiya jarayonidan so'ng kirish mumkin.

- Tizim bu yagona identifikatorlardan, foydalanuvchilar harakatini kuzatish uchun, foydalanish xolatida bo'lishi kerak.

- OT ob'ektlarni qayta foydalanishdan ximoya qilishi kerak. Yangi foydalanuvchiga hamma ob'ektlar, xotira va fayllarni ham, ajratishdan oldin initsiallashtirilishi kerak.

- Tizimli administrator, himoyaga mansub bo'lgan hamma xodisalarning hammasini hisobini olib borish imkoniga ega bo'lishi kerak.

- Tizim o'zini tashqi ta'sir va yuklangan tizimda va diskda saqlanadigan tizimli fayllarni o'zgartirish kabi xolatlardan ximoya qila olishi kerak.

Hozirgi kunda — “Oranjevaya kniga” o'rniga Common Criteria standarti keldi, va S2 sinf ko'rsatkichlarini (kriteriyani) Controlled Access Protection Profile ko'rsatkichlar majmuasi almashtirdi. Bu xujjatlarda axborot xavfsizligi bilan bog'liq asosiy tushunchalar ta'riflari berilgan.

Moxiyati jixatdan, xavfsizlik tizimni loyihalash, quyidagi savollarga javob berishni ko'zda tutadi: qanday ma'lumotlarni himoyalash, tizim xavfsizligiga qanday xujumlar taxdid qilishi mumkin va ma'lumotlarning har bir ko'rinishini ximoyalash uchun qanday vositalardan foydalanish mumkin. Bu savollarga javobni qidirish o'z chiga texnik aspektlardan tashqari, tashkiliy muammolarni ochishni ilgan xavfsizlik siyosatini shakllantirish deb ataladi.

Xavfsizlik siyosatini shakllantirishda quyidagi asosiy printsiplarni sobga olish zarur: m-n, Zaltser va SHreder (1975) MULTICS bilan

ishlash borasida tajribalariga asoslanib, OT larni loyihalashda ularning xavfsizligini hisobga olish uchun quyidagi tavsiyalarni ishlab chiqdilar.

- Tizimni loyihalash ochiq bo'lishi kerak. Buzg'unchi shundoq ham hamma narsani biladi. (kriptografik algoritmlar ochiqdir)

- Standart bo'yicha (po umolchaniyu) murojaat (dostup) bo'lmasligi kerak. Mualliflashtirilmagan murojaat ruxsat etilgan joydagi xatodan ko'ra — “yangi tizim” murojaat rad etiladigan joydagi xato tez topiladi.

- Joriy mualliflik sinchiklab tekshirilishi kerak. Ko'p tizimlar tekshirmaydilar. Natijada foydalanuvchi faylni ochishi va xafta davomida fayl egasi ximoyani o'zgartirsa ham uni ochiq holatida tutib undan foydalanishi mumkin.

- Har bir jarayonga imtiyozlarni mumkin qadar kam berish kerak.
- Ximoya mexanizmlari sodda, doimiy va tizim quyi qatlamlarida joylashgan bo'lishi kerak.
- Fizik jihatdan qo'llash imkoniyati muximdir. Foydalanuvchi, Xujumdan kelib chiqqan zarar va uni oldini olishga ketadigan xarajat balanslashtirilgan bo'lishi kerak.

YUqorida keltirilgan fikrlar ximoya mexanizmini tizimni loyixalashning eng boshlanishida hisobga olinishini taqazo etadi.

13.3. Tarmoq tizimlarining auditi

Eng yaxshi ximoya ham ertami-kechmi buziladi, shuning uchun ham, buzilishlarga bo'lgan xarakterlarni aniqlash, ximoya tizimining muhim masalasi bo'lib qoladi, chunki bu masala echimi buzilishlardan kelib chiqqan zararlarni kamaytirish va buzish usullari haqida ma'lumot yig'ishga imkon beradi. Ko'pincha, albatta buzg'unchilar legal foydalanuvchidan o'zini tutishi bilan farq qiladi. Ba'zan bu farqlarni sonli ravishda aniqlash mumkin bo'ladi, m-n, parolni noto'g'ri kiritish xollari soni.

Audit, shunday qilib, tizimda ro'y beradigan har turdagi xodisalarni ro'yxatga olishdan iboratdir. Bu xodislar u yoki bu xolatda kompyuter tizimi xavfsizligi xolatiga ta'sir etadi. Bunday xodisilarga odatda quyidagilar kiradi:

- tizimga kirish yoki chiqish;
- fayl ustida amallar (ochish, yopish, qayta nomlash, olib tashlash);
- olib tashlangan tizimga murojaat xavfsizlik atributlari (murojaat rejimi, foydalanuvchi ishonchlilik darajasi va x.k.) yoki imtiyozlar o'zgarishi.

Agar hamma hodisalar ro'yxatga olinaversa, ro'yxatga olingan ma'lumotlar xajmi juda tez ko'payib ketadi, va uni samarali taxlil qilish iloji bo'lmay qoladi. SHuning uchun foydalanuvchiga nisbatan ham (shubhali shaxslar kuzatilayapganda), xodisalarga nisbatan ham tanlangan protokollashtirish vositalari mavjudligini nazarda tutish lozim.

Protokollashtirishdan tashqari vaqti-vaqti bilan tizimni skanerlab turish mumkin, bunda xavfsizlik tizimidagi nozik joylar qidiriladi.

- qisqa yoki oson parollar;
- mualliflashshtirilmagan dasturlar;
- tizimli direktoriyalardagi mualliflashtirilmagan dasturlar;
- uzoq bajariladigan dasturlar;
- mantiqiy bo'lmagan ximoya (foydalanuvchi, tizimli direktoriy va fayllar);
- tizimli dasturlardagi o'zgarishlar va x.k.

Xavfsizlik skaneri yordamida topilgan ixtiyoriy muammo avtomatik tarzda echilishi yoki tizim menedjeriga xal qilish uchun berilishi mumkin.

Ba'zi OT larni ularning ximoyalanganligi nuqtai-nazaridan taxlili.

OT xavfsizlik choralarini amalga oshirishga yordam berishi yoki ularni qo'llashi kerak. Apparatura doirasida va OT doirasidagi echimlarga quyidagilar misol bo'la oladi:

- komandalar imtiyozligi darajasi bo'yicha ajratish;
- jarayonlarni adres makonlarini segmentlash va segmentlar imoyasini tashkil etish;
- turli jarayonlarni bir-birlariga o'zaro ta'siridan, har biriga o'z tual makonini ajratish hisobiga himoya qilish;
- OT yadrosini alohid ximoyasi;
- ob'ektlardan qayta foydalanishdan nazorat qilish;
- murojaatni boshqarish vositalari mavjudligi;
- tizimning strukturalashtirilganligi, ishonchli ximoya;

- ishonchli hisoblash bazasini ajratish (ximoyalangan komponentalar yig'indisini) bu baza ixchamligini ta'minlash;
 - imtiyozlarni minimallashtirish printsipligini ta'minlash;
 - komponentalarga, ular funksiyalarni ta'minlashga qancha imtiyoz kerak bo'lsa, shuncha imtiyoz beriladi.
- Fayl tizimi strukturasi katta ahamiyatga egadir. Umuman, xavfsizlik charalari, odindan OT ga kiritilgan bo'lishi shart emas, ximoya maxsulotlarini qo'shimcha o'rnatish imkoniyati yetarli.

13.4. Kompyuterlarning xavfsizlik darajalari

Ko'pgina axborot xavfsizligi xizmatlari, masalan, tizimga kirishi nazorat, resurslarga murojaat chegaralash, ma'lumotlarni saqlash xavfsizligini ta'minlash va qator xavfsizlik xizmatlari kriptografik algoritmlardan foydalanishni nazarda tutadi.

Shifrlash – bu shunday jarayonki, unda ochiq tekst (plaintext) ma'lumotlari, shifrotekstga (cliphertext) shunday tarzda o'tkaziladi:

- uni faqat, kim uchun tayinlangan bo'lsa faqat u o'qiy oladi.
- jo'natuvchini haqiqiylikini tekshiriladi (autentifikatsiya).
- Jo'natuvchi haqiqatdan ham ayni ma'lumotni jo'natganligi kafolatlanadi.

SHifrlash algoritmlarida albatta kalit mavjud bo'ladi. SHifrlashning maxfiy, simmetrik, ochiq va assimetrik va x.k. usullari mavjud. SHifrlashda turli matematik funktsiya va algoritmlardan foydalaniladi.

OT xavfsizligi masalalarini yechish, ularning arxitektura xususiyatlari va identifikatsiya va autentifikatsiyani mualliflashtirish va auditni to'g'ri tashkil qilish bilan bog'liqdir.

OT larning ximoyasi asosiy masalalari identifikatsiya, autentifikatsiya, foydalanuvchilarning resurslarga murojaat xuquqlarini chegaralash, protokollashtirish va tizim auditi kiradi.

Tizimga kirish (murojaat) kirish nazorat muammosini ko'rib chiqamiz. Eng tarqalgan nazorat usuli bu ro'yxatga olish protsedurasidir. Odatda har bir foydalanuvchi tizimda o'z yagona identifikatoriga egadir. Foydalanuvchi identifikatorlari, ixtiyoriy boshqa ob'ektlar, fayllar va jarayonlar identifikatorlariga o'xshas maqsadlarda foydalaniladi. Identifikatsiya qilish, foydalanuvchi

tomonidan o'z identifikatorini xabar qilishdir. Foydalanuvchi haqiqatan ham xaqiqiy va kiritilgan identifikator uniki ekanligini aniqlash uchun axborot tizimlarida autentifikatsiya (authentication) protsedurasi ko'zda tutiladi. Bu so'z lotinchadan tarjimasi — "xaqiqiylikni aniqlash"ni bildiradi va bu jarayon maqsadi tizimga mumkin bo'lmagan shaxslarni kiritmaslikdir.

Odatda autentifikatsiya quyidagi bo'limlarning biri yoki bir nechtaga asoslanadi:

- foydalanuvchi bor narsa (kalit yoki magnet kartasi);
- foydalanuvchi biladigan narsa (parol);
- foydalanuvchi atributlari (barmoq izlari, imzo tovush).

Parollar va ularning kamchiligi

Autentifikatsiyaning eng oson yonlashishi-foydalanuvchi parolini qo'llashdir.

Foydalanuvchi o'zini yagona identifikator yoki nomi yordamida autentifikatsiya qilsa, undan parol so'raladi. Agar foydalanuvchi bergan parol tizimda saqlanayapgan parol bilan mos tushsa, tizim foydalanuvchi — "legitimen" deb hisoblaydi.

Kompyuter tizimida ob'ektlarni ximoya qilish uchun ko'pincha, murakkab ximoya tizimlari bo'lmagan xollarda, parollar ishlatiladi.

Parollar kamchiligi, foydalanuvchi uchun parol qulayligi va uning ishonchliligi orasida balans saqlash qiyinligidir. Parolni topish, tasodifan ko'rsatib qo'yish va yashirin tarzda mualliflashmagan foydalanuvchiga berilishi mumkin. Bundan tashqari talaygina parollarni topishga urinish usullarini keltirishimiz mumkin.

SHunga qaramasdan, parollar keng tarqalgan, chunki ular qulay va engil amalga oshiriladi.

Parolni shifrlash.

Parollar maxfiy ro'yxatini diskda saqlash uchun, ko'pgina OT larda, kriptografiyadan foydalaniladi.

Kodlashtirilgan parollargina saqlanadi. Autentifikatsiya jarayonida, foydalanuvchi bergan parol kodlashtiriladi va diskdagi bilan aqqoslanadi. SHunday qilib, parollar faylini yashirin holda saqlash aruriyati yo'qdir.

Avtomatlashtirish (mualliflashtirish). Ot ob'ektlariga murojaatni egaralash.

Muvaffaqiyatli ro'yxatdan o'tgandan so'ng, tizim avtorlashtirishni amalga oshiradi, ya'ni sub'ektga ob'ektga murojaat xuquqini berishni amalga oshiradi. Avtorlashtirish vositalari legal foydalanuvchilarga, ularga administrator belgilagan xuquqlar bo'yicha, tizim resurslariga murojaatini nazorat qiladi, va shu bilan birga foydalanuvchilarga turli tizimli funksiyalarni bajarish imkonini beradi. Nazorat tizimi —murojaat matritsasil (matritsa dostupa) nomli umumiy modelga asoslanadi.

Murojaatni boshqarishning —diskretсионno'yl (saylash) va "polnomochno'y" (mandatli) usullari mavjud.

13.5. Nazorat savollari

1. Axborot xavfsizligini ta'minlash uchun qanday usul vositalar talab qilinadi?
2. Xavfsiz tizimlar xossalari.
3. Tizim ishini buzuvchi dastur va tizim xavfsizligiga taxdidlar.
4. OT ni loyihalash asosiy printsiplari (MULTICS) va xavfsizlik sinflari talablari (S2).
5. Kriptografiyadan xavfsizlik uchun foydalanish.
6. Identifikatsiya va autentifikatsiya.
7. OT ob'ektlariga murojaatni chegarasi, avtorlashtirish. Xavfsizlik tizimi auditi.
8. Ommaviy OT larni, ximoyalanganlik nuqtai nazardan taxli qilish.

14. BULUTLI HISOBLASH (CLOUD COMPUTING) UCHUN OPERATSION TIZIMLAR

14.1. Cloud computing tushunchasi

Bulutli hisoblashlar va virtuallashtirish texnologiyasi hozirgi paytda juda faol tadqiqot sohalari hisoblanadi. Bu sohalarda sanoqsiz tadqiqot ishlari amalga oshirilyapti. Xar bir sohada bir nechta ilmiy tadqiqot anjumanlar o'tkaziladi. Masalan, hisoblash texnikasi assosiyatsiyasi tomonidan o'tkaziladigan bulutli hisoblashlar bo'yicha simpozium (ACM Symposium on Cloud Computing (SOCC)) bulutli hisoblashlar bo'yicha masalalar ko'riladigan mashxur maydonchadir. SOCC doirasida chop qilinadigan maqolalar ma'lumot markazlari masalalari, bulutlarni o'rnatish va otladka qilish va h.k. kabi muammolarni o'z ichiga oladi.

Virtuallashtirish texnologiyasi bulutli hisoblashlar sohasi rivojlanishida muhim o'rin egallaydi. Bulutli texnologiyalarning bir nechta turlari mavjud. Ba'zilari pullik asosda xamma uchun ochiq bo'lsa, ba'zilari biror korxonaning maxsus berk buluti bo'lishi mumkin. Shuningdek turli bulutlar turlicha xizmatlarni taklif qiladilar. Ba'zilari o'zining foydalanuvchilariga fizik qurilmalarini taqdim qilsalar, ba'zilari dasturiy muhitini taqdim qiladilar holos. Bulutli xizmatlar ishlab chiqaruvchilar turli kategoriyadagi resurslarni taklif qiladilar, masalan, "katta mashinalar", "kichik mashinalar" va h.k.

VMWARE mahsulotlari

1999 yildan boshlab VMware, Inc. virtualashtirish sohasida yetakchi ishlab chiqaruvchi korxonalaridan biri hisoblanadi. Korxonada shaxsiy kompyuterlar, serverlar bulutlar va uyali telefonlar uchun mahsulotlar ishlab chiqaradi. Kompaniya na faqat gipervizorlarni, balki katta masshtabdagi virtual mashinalarni boshqaruvchi dasturlarni xam ishlab chiqaradi. Gipervisorlar – bu bir kompyuterda bir nechta operatsion tizimni parallel bir vaqtda ishlatish imkonini beruvchi dastur ki qurilma.

Virtual mashinalardan foydalanish g'oyasi 1960-1970 yillarda ham kompyuter sanoatida, ham akademik tadqiqotlarda ommalashgan bo'lsa-da, 80-yillardan keyin virtualizatsiyaga bo'lgan qiziqish

butunlay yo'qoldi va shaxsiy kompyuterlar ishlab chiqarish tobora ortib bordi. Faqatgina IBMning Universal Machines bo'limi virtualizatsiya bilan shug'ullanishni davom etdi. Darhaqiqat, o'sha paytdagi kompyuter arxitekturalari, xususan Intelning x86 arxitekturasi, virtuallashtirishni qo'llab quvvatlamagan.

1997 yilda Stenfordda "VMware"ning kelajakdagi uchta asoschilari Disko nomli gipervizor prototipini yaratadilar. Bu FLASH mashinasida, Stenfordda ishlab chiqarilgan juda katta mikroprotsessorda tijorat operatsion tizimlarini (xususan UNIX) boshqarish uchun ishlab chiqiladi. Loyihani yaratish mobaynida mualliflar tushunishdiki, virtual mashinalardan foydalanish bir vaqtning o'zida bir operatsion tizimning necha murakkab masalalarni bajarish mumkin ekan. Natijada 1998 yilda VMware, Inc. kompaniyasiga asos solindi va x86 arxitekturasi xamda shaxsiy kompyuterlar sanoatiga virtuallashtirish funksiyasi kiritildi. VMware-ning birinchi mahsuloti -VMware Workstation 32-bitli x86 asosidagi platformada mavjud bo'lgan birinchi virtuallashtirish yechimi edi. Mahsulotning birinchi chiqarilishi 1998 yilda ikkita variantda taqdim qilindi: Linux uchun VMware Workstation, bu asosiy Linux operatsion tizimining ustida ishlaydigan 2-tur gipervizor va Windows NT-ning ustida ishlaydigan Windows uchun VMware Workstation. Ikkala variant ham bir xil funktsiyaga ega edi: foydalanuvchi virtual apparatning xususiyatlarini oldindan belgilab, bir nechta virtual mashinalarni yaratishi mumkin (masalan, virtual mashinaga qancha xotira berish yoki virtual disk hajmini qanday aniqlash).

AQShning Milliy standartlar va texnologiyalar instituti bulutli texnologiyalarga tegishli quyidagi 5 ta xususiyatlarni sanab o'tadi:

1. Talab bo'lganda o'z-o'ziga xizmat qilish (On-demand self-service). Foydalanuvchilar inson faktori aralashuvisiz resurslardan avtomatik ravishda foydalana olish imkoniga ega bo'lishlari kerak.
2. Tarmoqdan keng foydalanish (Broad network access). Barcha resurslarga standart mexanizmlar yordamida tarmoqdan murojaatlilik ta'minlanishi kerak.
3. Resurslarni birlashtirish (Resource pooling). Taqdim qilinadigan kompyuter resurslari bir nechta foydalanuvchilarga taqdim qilinishi uchun dinamik tarzda biriktirib va ozod qilish uchun umumlashtirilishi

lozim. Foydalanuvchilar odatda "o'zlarining" resurslari qayerda va xattoki qaysi mamlakatda joylashganligini bilishmaydi xam.

4. Tezkor elastiklik (Rapid elasticity). Resurslar ixtiyoriy paytda avtomatik ravishda taqdim qilinishi yoki ozod qilinish imkoni bo'lishi kerak.

5. Inobatga olingan xizmatlar. (Measured service). Resurslarni taqdim qiluvchi korxonalar ishlatilgan resurslarni hisobini yuritishi lozim.

14.2. Xizmatlar ma'lumotlarni qayta ishlash markazlari

14.2.1. Xizmat sifatida infratuzilma (IaaS)

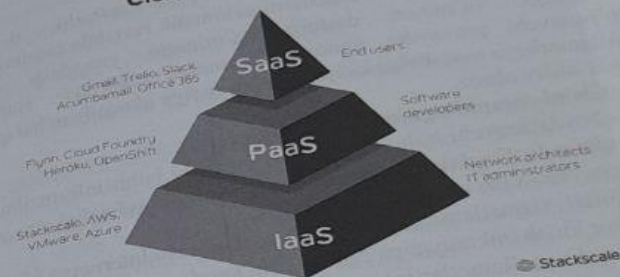
Ushbu model asosida iste'molchi axborot texnologiyalari resurslarini - ma'lum hisoblash quvvati va xotira hajmiga ega virtual serverlarni oladi. Provyayder barcha apparat ta'minot bilan shug'ullanadi. U virtual mashinalarni yaratish uchun dasturlarni o'rnatadi, lekin foydalanuvchi dasturlarini o'rnatmaydi va saqlamaydi. Provyayder faqat haqiqiy va virtual infratuzilmani boshqaradi. IaaS misollari: IBM Softlayer, Hetzner Cloud, Microsoft Azure, Amazon EC2, GigaCloud. IaaS mijozlari - bu kompaniyalarning tizim ma'murlari.

14.2.2. Xizmat sifatida platforma (PaaS)

Bunday holda, bulut provayderi operatsion tizimlarga, ishlab chiqish va sinov vositalariga va ma'lumotlar bazasini boshqarish tizimlariga kirishni ta'minlaydi. Provyayder nafaqat serverlarni, saqlash tizimlarini va hisoblash quvvatini boshqaradi, balki foydalanuvchiga ularni boshqarish uchun ma'lum platformalar va vositalarni tanlashni ham taklif qiladi. PaaS misollari: Google App Engine, IBM Bluemix, Microsoft Azure, VMWare Cloud Foundry. PaaS xizmatidan foydalanuvchilar - bu dasturiy ta'minot ishlab chiqaruvchilardir.

14.2.3. Dastur xizmat sifatida (SaaS)

Cloud service models



Ushbu bulut modeli eng keng tarqalgan. Dasturlar va xizmatlar provayder tomonidan ishlab chiqiladi va saqlanadi, bulutga joylashtiriladi va oxirgi foydalanuvchiga brauzer yoki uning shaxsiy kompyuteridagi dastur orqali taqdim etiladi. Mijoz faqat oylik to'lovni to'laydi (yoki xizmatdan bepul foydalanadi), provayder dasturlarni yangilash va texnik qo'llab-quvvatlash uchun javobgardir. SaaS xizmatlari saqlash uchun joy (Dropbox), ish uchun ofis hujjatlari to'plami (Google Doc, Microsoft Office 365), fotosuratlarini tartibga solishda (Flickr) yoki boshqalar bilan bog'lanishda (Facebook) yordam berishi mumkin. SaaS xizmatlarining asosiy mijozlari - bu doimiy foydalanuvchidir.

14.3. Cloud computing uchun operatsion tizimlar va dasturiy vositalar obzori

Cloud computingning afzalliklari:

- Foydalanuvchi xizmatni faqat unga kerak bo'lganda to'laydi va eng muhimi, u faqat foydalangan narsasi uchun to'laydi.

- Bulutli texnologiyalar dasturiy ta'minot va uskunalarni sotib olish, qo'llab-quvvatlash, modernizatsiya qilishda tejashga imkon beradi.

- Mashtablilik, xatolarga chidamlilik va xavfsizlik - ilovaning ehtiyojlariga qarab zarur resurslarni avtomatik ravishda taqsimlash va bo'shatish. Ta'minot, dasturiy ta'minotni yangilash xizmat ko'rsatuvchi provayder tomonidan amalga oshiriladi. Bulutdagi ma'lumotlarga masofadan kirish - siz Internetga ulanadigan har qanday joyda ishlay olasiz.

Cloud computingning kamchiliklari

- Foydalanuvchi bulut egasi emas va ichki bulut infratuzilmasiga kira olmaydi. Foydalanuvchi ma'lumotlarining xavfsizligi provayder kompaniyasiga uzviy bog'liq.

- Sifatli xizmatlarni olish uchun foydalanuvchi Internetga ishonchli va tezkor kirish imkoniyatiga ega bo'lishi kerak.

- Barcha ma'lumotlar Internet-provayderga nafaqat saqlash uchun, balki uni qayta ishlash uchun ham ishonib topshirilishi mumkin.

- Onlayn xizmat ko'rsatuvchi provayder bir kun zaxira nusxasini yaratmaslik xavfi mavjud va ular server ishdan chiqishi natijasida yo'q bo'lishi mumkin.

- Ma'lumotlaringizni onlayn xizmatga ishonib, siz ular ustidan nazoratni yo'qotasiz va erkinligingizni cheklaysiz.

14.3.1. Bulutli dastur

Ta'limda bulutli texnologiyalardan foydalanishning namunasi sifatida elektron kundaliklar va jurnallar, talabalar va o'qituvchilarning shaxsiy hisoblari, interfaol qabulxona va boshqalarni chaqirish mumkin. Bular talabalar ma'lumot almashishlari mumkin bo'lgan tematik forumlardir. Bu o'quvchilar o'qituvchining yo'qligida yoki uning rahbarligi ostida ham ma'lumotni ta'lim muammolarini hal qilishlari mumkin bo'lgan ma'lumotni izlashdir. Buning uchun quyidagilardan foydalanishingiz mumkin:

- kompyuter dasturlari
- elektron darsliklar
- fitnes jihozlari
- diagnostika, sinov va o'qitish tizimlari

- amaliy va asbobli dasturlar
- laboratoriya majmualari
- multimedia texnologiyasiga asoslangan tizimlar
- telekommunikatsiya tizimlari (masalan, elektron pochta, yangiliklar guruhlari)
- elektron kutubxonalar va boshqalar.

14.3.2. Ta'lim uchun Microsoft buluti

Microsoft Cloud Technologies: Office 365, Ta'lim uchun Azure Microsoft Office 365 ta'lim muassasalari uchun "bulutli" xizmatlarning barcha imkoniyatlaridan foydalanishga imkon beradi, bu vaqt va pulni tejashga yordam beradi, shuningdek talabalar va xodimlarning ishlash samaradorligini oshiradi. Exchange Online, SharePoint Online va Office veb-ilovalarining bulutli versiyalari, shuningdek, videokonferentsaloqa bilan Lync Online-ning asosiy funksiyalari bepul taqdim etiladi.

Ta'limda Windows Azure-dan foydalangan holda o'qituvchilarga o'quv jarayoniga nazariy va amaliy jihatdan eng innovatsion va tez rivojlanayotgan texnologiyalarni kiritish imkoniyati beriladi. Ta'lim universitetlarida ishqibozlik bunday mutaxassislarni tayyorlash imkoniyatiga ega.

Bulutli hisob-kitoblar dasturlarni ishga tushirishni yoki Internet orqali ulanadigan ma'lumotlar uzatish markazlarida joylashgan serverlarda ma'lumotlarni saqlashni o'z ichiga oladi. Bunday dasturlarni ishlab chiqish va ishlatish uchun maxsus bulutli platforma kerak. Bunday platforma Windows Azure, Windows Server operatsion tizimining "bulutli" analogidir. Ammo, agar Windows Server siz o'zingizning mahalliy ma'lumotlar markazingizdagi serverlarda sotib olgan va o'rnatgan dasturiy ta'minot bo'lsa, unda Windows Azure platformasi Microsoft ma'lumotlar markazlarida joylashgan va sizga masofadan turib dasturlarni ishlab chiqish va ishlatish uchun muhim sifatida foydalanish mumkin. Sizga dasturiy ta'minotni sotib olish va o'rnatish shart emas, siz faqat hisoblash manbalarini ijarasi Microsoft ma'lumotlar markazi platformasi sig'imi uchun to'laysiz.

Windows Azure dasturning har bir misoli uchun virtual mashinani ishga tushirishga asoslangan. Dasturchi kerakli miqdor

ma'lumotlarni saqlash va kerakli hisoblash quvvatini (virtual mashinalar soni) aniqlaydi, shundan so'ng platforma (virtual manbalarni taqdim etadi. Dastlabki resurs talablari o'zgarganda, mijozning yangi so'roviga binoan, platforma qo'shimcha ma'lumotlarga ega yoki dastur uchun foydalanilmagan ma'lumotlar markazlarini kamaytiradi.

14.4. Nazorat savollari

1. Bulutli hisoblash uchun operatsion tizimlar
2. Virtuallashtirish va bulutli texnologiyalar
3. Bulutlar
4. Uch darajali bulutli xizmatlar.
5. Xizmat sifatida infratuzilma (IaaS)
6. Xizmat sifatida platforma (PaaS)
7. Dastur xizmat sifatida (SaaS)
8. Bulutli hisoblashning afzalliklari
9. Bulutli hisoblashning kamchiliklari
10. Bulutli dastur
11. Ta'lim uchun Microsoft buluti

15. OPERATSION TIZIM VA TARMOQLARNING ISTIQBOLI

15.1. IBM, HP, Oracle/Sun va boshqa firmalarning Operatsion tizimlari

IBM PC shaxsiy kompyuterni Microsoft firmasining MS DOS operatsion tizimi yoki uning IBM firmasi tarqatadigan PC DOS varianti rahbarligida yo'bo'lmasa, Digital Research firmasi (hozirgi Novel firmasining bo'linmasi)ning MS DOS operatsion tizimi bilan qo'shilib, o'rin almashib ishlay oladigan DR DOS tizimi yoki IBM firmasining PC DOS operatsion tizimi boshqaruvida ishlaydilar. Bundan buyon bu uchta operatsion tizimi ta'riflanadi va bunda ularning hammasi bitta umumiy DOS so'zi bilan ataladi.

Dasturiy interfeys – xisoblash tizimi doirasida qurilma va dasturlar o'zaro ta'sirini ta'minlovchi vositalar yig'indisidir. Foydalaniluvchi interfeys- foydalanuvchining dasturiy yoki EXM bilan uzaro ta'siridagi dasturiy va apparat vositalaridir. Xar bir kompyuter albatta operatsion tizim turkumiga ega buladi, ularning xar biri uchun operatsion tizim turkumi yaratiladi. Operatsion tizimlar quyidagi omillar bo'yicha tavsiflanadi:

Bir vaqtda ishlaydigan foydalanuvchilar: bir (kishi) foydalanuvchi, ko'p (kishi) foydalanuvchilar soniga ko'ra;

Tizim boshqaruvi ostida bir vaqtda foydalaniluvchi jarayonlar: bir vazifali, ko'p vazifalilar soniga ko'ra;

Qo'llab-quvvatlovchi jarayonlar: bir protsessorli, ko'p protsessorlilar soniga ko'ra;

OT kodi razryadlilikigi: 8 razrayadli, 16 razrayadli, 32 razrayadli, 64 razrayadlilikiga ko'ra;

Interfeys turi: buyruq (matnli) va ob'ektili-yunaltirilgan (grafik)ligiga ko'ra;

Foydalaniluvchining EXMga kirishi turi: paketli qayta ishlash, va bilan bo'linishi, real vaqtga ko'ra;

Zaxiralardan foydalanish turi: tarmokli, lokalliligiga ko'ra.

Hozirgi paytda operatsion tizimlarning DOS; OS/2; UN Windows oilalari keng tarkalgan. DOS oilasidagi operatsion tizimlarning birinchi vakili-MS DOS tizimi 1981 yilda IBM PC p

bo'lishi munosabati bilan chikarilgan. DOS oilasining operatsion tizimlari bir vazifali bo'lib, kuyidagi uziga xos xususiyatlarga ega:

EXMli interfeys foydalanuvchi kiritadigan buyruk yordamida amalga oshiriladi; Tizimning EXMning boshqa turlariga utishini soddalashtiradigan tuzilma mavjudligi;

Operativ xotiraga kirish xajmining uncha katta emasligi (640 Kbayt);

DOS operatsion tizim oilalarining jiddiy kamchiligi SHK va OT zaxiralarga beruxsat kirishdan muxofaza vositalarining yo'qligidir.

Windows operatsion tizimining umumiy ta'rifi.

Windows oilasidagi OT lar Microsoft firmasi tomonidan tayyorlangan. Ular qulay grafik interfeysni uzida namoyon etuvchi ko'p vazifali operatsion tizimlardir. Windows 95 operatsion tizim va Windows NT OT lar shu oilaning asosiy vakilidir. Windows 95 tizimi va yangi to'laqonli operatsion tizim bo'lib, qudratli va foydalanishda oddiydir. Agar Windows 3.x ning ilgari versiyalari MS DOS ni asos sifatida olgan bo'lsa, Windows 95 esa kompyuterda biror bir boshqa operatsion tizim ishtirokini talab etmaydi. Siz mashinaga bittagina maxsulotni o'rnatasiz va darhol. Windows muxitiga tushasiz. Bundan tashkari. Windows 95 shunchaki operatsion tizimgina emas. Bu ko'plab yangi, foydali va qiziqarli narsalarni uzida saklaydigan dasturiy maxsulotdir. Oddiy va tushunarli xujjatga yunaltilgan interfeys, kontekstli menyu, sozlashning oddiyliigi, DOS ilovalar va Windows ning oldingi versiyalari bilan tulik moslik, shuningdek, umuman butun tizimning g'aroyib kuch kudrati – uzoq va mashaqqatli ishi natijasi bo'ldi. Endi hatto yangi foydalanuvchi ham mazkur yangi OT bilan ishlashda o'zini ishonchsiz his etmaydi. Foydalanuvchi qanday vaziyatda qolmasin qulay va tushunarli interfeysda adashib qolmaydi. Ish jarayonida yordam beradigan va kompyuter texnikasi soxasidan uzoq bo'lgan kishiga mo'ljallangan interfeys bu mahsulotni ommaviy iste'molchiga qulay qilib qo'ydi.

Windows 95 OTning asosiy menyusida ishlash juda ham oddiy bo'lib, foydalanuvchi uchun qo'shimcha qiyinchiliklar tug'dirmaydi. Microsoft kompaniyasining boshqa istalgan maxsuloti kabi Windows 95 ni ham uchta oddiy so'z bilan izoxlash mumkin: oddiylik, kuvvatlik, moslik.

Windows operatsion muxiti quyidagi imkoniyatlarga ega:

Universal grafika- Windows dasturlarining qurilmalariga va ta'minotga bog'liqligini ta'minlaydi.

Yagona interfeys- Windowsda foydalanuvchining muloqoti uchun yangi dastur bilan ishlash qoidalari umumiy. SHuning foydalanishingiz mumkin.

Mavjud dastur ta'minot bilan muvofikligi – Windowsning barcha paketlari taxrirlagichlari, elektron jadvallari ishini to'la amalga oshiradi.

Ko'p masalaligi – Windows bir paytning o'zida bir necha masalani hisoblaydi, bir dasturdan boshqasiga o'tishni ta'minlaydi.

Mavjud operativ xotirada to'la foydalana oladi, qurilma resurslaridan ham to'liq foydalanadi.

Ma'lumotlar almashuvi – Windows dasturlararo ma'lumotlar almashish imkoniyatiga ega. Bu maxsus Clipboard (ma'lumotlar buferi) yoki DDE (ma'lumotlarning dinamik almashuvi, ya'ni boshqa dastur natijalaridan foydalanish) yoki OLE (ma'lumotlardan ularni taxrirlagan holda foydalanish) yordamida amalga oshiriladi.

Windowsda ekran "deskto" deb ataladi. Barcha amallar unda bajariladi. Windows so'zining ma'nosi "oynalar" (darchalar). SHuning uchun Windowsga kirganda dasturlar darchalarda bajariladi. Darchalarni joyini va o'lchamlarini uzgartirish mumkin.

Agar bir nechta dastur bajarilayotgan bo'lsa, darchalarni piktogrammalarga kichraytirish mumkin. Bu esa ekrandan unumli foydalanish imkonini beradi.

Darchalar uch xil bo'ladi:
Dasturlar darchasi;
Xujjatlar darchasi;
Muloqot darchasi.

Darcha yoki piktogramma joyini o'zgartirish uchun sarlavxani sichkoncha tugmachasini bosgan xolda yangi joyga keltiriladi.

Windows dasturi yuklanganda uning ishchi stolida "moy kompyuter", "moi dokument", "portfel", "korzina" piktogrammalari (istalgan piktogrammalarni joylashtirish mumkin) namoyon bo'ladi.

Ekraning pastida joylashgan "Panel zadach"da "Pusk" tugmasi joylashgan bo'lib, unda Bosh menyu joylashgan.

OT interfeysi asosiy elementlari boshlanishiga Windows deb atalmish chiroyli binoning asosiy elementar bo'laklarini sanab o'tamiz. Ekraning asosiy qismini "Rabochiy stol" (ish stoli) egallagan. Metafora (o'xshashlik) tushunarli: chiroyli rasmlar bilan bezatilgan makonda biz ish qurollarimiz - xujjatlar, papkalar va kerakli dastur belgilarini va h.k ni joylashtiramiz. Biz ularni - xuddi ofis stolida ruchka, skrepka va turli ish qurollari bo'lgani kabi joylashtiramiz. Ishchi stol. Ish stolining quyi qismida yarim shaffof lenta - masalalar panelini (panel zadach) ko'ramiz. Bu yerda eng zarur va kerakli dastur belgilari joylashgandir. Bu panelda ba'zi belgilar oldindan mavjud, qolganlarini o'zimiz o'rganamiz. (Ixtiyoriy belgini sichqoncha yordamida bu panelga o'tkazish mumkin). Masalalar paneli chap burchagida "Pusk" tugmasi joylashgan. Uning yordamida siz kompyuterga o'rnatilgan hamma dasturlar ro'yxati va sozlashning asosiy quroliga murojat qilishingiz mumkin.

O'ng burchakda yana bitta panel joylashgan, bu tizimli panel deb ataladi. Bu erda ham ishga tushirilgan dasturlar belgilariga ko'rsatkichlar aks ettirilgan bo'lib, farq shundaki, ular sizning kompyuteringizda oson rejimda doimo ishlab turadi.

Bugungi kunga kelib, axborot hajmining keskin ortib borishi va talab qilinayotgan vazifalar turining ko'pligi hisoblash texnikasiga bo'lgan talabning ham ortishiga olib keldi. Hisoblash texnikasi hayotimizning turli sohalarida (ilmiy masalalar, matematik modellash, ma'lumotlar bazasi, Internet server, xosting, server prilojeniy va boshqalar) keng qo'llanilmoqda. Bular albatta - serverlardir.

Serverlarning turi kundan kunga ko'payib bormoqda va unga mo'ljallangan ehtiyot qismlarning ham soni oshib bormoqda. Agar sizning kompaniyangiz uzoq yillar davomida uzilishsiz ishlashini istasangiz o'z navbatida serverlarning eng yaxshisini va zamonaviysini tanlaganingiz ma'qul. Bu albatta katta mablag' talab qiladi. (pulingiz yetadimi?).

Blade - lezviya degani. Bu XXI asrning serverlaridir. Bladening birinchi nusxasi 2001 yilda ishlab chiqarilgan. Ushbu serverlar HP, IBM, Fujitsu va boshqa kompaniyalar tomonidan ishlab chiqariladi. Agar sizda HP firmasining shassisi bo'lsa siz IBM firmasining blade-serveridan foydalana olmaysiz.

MySQL bu - krossplatformali dastur hisoblanadi, ya'ni bir xil tipga ega bo'lgan (ishlash texnologiyasi bir xil), lekin o'rnatiluvchi fayllari biron boshqachadir. Hozirda mysql Oracle firmasi tomonidan ishlab chiqarilmoqda, oldin Sun firmasiga tegishli bo'lgan. Shuning uchun MySQL 4talqini MySQL 5 talqinidan farq qiladi. Hozirda (maqola yozilishida) eng so'ngi talqini 5.6.14 dir.

MySQL juda ko'p operatsion tizimlar bilan ishlay oladi. Bularni yozadigan bo'lsam: AIX, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetBSD, OpenBSD, OS/2 Warp, SGI IRIX, Solaris, SunOS, UnixWare, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Server 2000, Windows Vista, Windows 7,.... MySQL shved korxonasi MySQL AB ga tegishli bo'lgan, oldin esa, gigant korxonalardan biri Oracle, SUN firmasini sotib oldi va hozirda MySQL Oracle firmasi mahsuloti sifatida chiqib kelmoqda. Bu bir hisobda Microsoft SQL Server ga katta raqobatdosh ekanligini anglatadi.

MySQL bir necha serverlarning bir qismi hisoblanadi. Misol uchun, WAMP, AppServ, LAMP, Denwer,.... Kliyentlar MySQL serveriga ma'lum bir kutubxonalar orqali ulanadi. MySQL ga quyidagi dasturlash tillari ulanib ishlab mumkin: Delphi, C, C++, Java, Perl, Php, Python, Ruby va boshqalar.

Bazaviy dasturiy ta'minlanish (base software)-kompyuterini ishlashini ta'minlovchi dasturiy vositalarning eng kichik majmuasidir. Servisli dasturiy ta'minlanish - bazaviy dasturiy ta'minlanishning imkoniyatlarini oshiruvchi va foydalanuvchiga qulayroq ish muhitini tashkil qiluvchi dasturlar va dasturiy majmualardir. Bazaviy dasturiy ta'minlashga quyidagilar kiradi:

- operatsion tizim;
- operatsion qobiqlar (matnli va jadvali);
- tarmoqli operatsion tizim.

Operatsion qobiqlar - foydalanuvchining operatsion tizimning buyruqlari bilan muloqatni engilashtirish uchun maxsus dasturlar. Operatsion qobiqlar yakuniy foydalanuvchi interfeysining matnli va jadvali variantlariga ega. Bu dasturlar operatsion tizim buyruqlarini bajarish uchun boshqaruvchi axborotlar vazifalarini ancha

soddalashtiradi, yakuniy foydalanuvchi ishining keskinligi va murakkabligini kamaytiradi.

15.2. Operatsion tizimlarning qobiqlari

Operatsion qobiqlar bevosita kushimcha dastur bo'lib har qanday operatsion tizim doirasida ma'lum-bir ishlarni yengillashtirish uchun mo'ljallangan. Ushbu dasturlar faylar va direktoriyalar bilan asosiy amallarni, ya'ni qidirish, nusxa ko'chirish, qayta nomlash, o'chirish kabilarni soddarak, qulayrok va yaqqol bajaradi.

Qobiq – biror bir dastur va foydalanuvchi urasidagi katlam yoki boshqa dastur ustida ustkurma bo'lgan dastur. Shunday qilib, operatsion qobiqlar operatsion tizimning vazifalarni bajaradi va uning imkoniyatlarini kengaytiradi.

Operatsion qobiqlar turli xil funksiyalarni bajaradi, xususan OT ning sozlovchisi bo'lgan qobiqlar operatsion qobiqlar deb ataladi.

Operatsion qobiqlar turli xil funksiyalarni bajaradi, xususan Diskdagi kataloglar ro'yxatini yaqqol ko'rsatadi;

Diskdagi kataloglar daraxtini ko'rsatish, kataloglarni yaratish, qayta nomlash, o'chirish imkoniyatlariga ega;

Fayllar ustida nusxa ko'chirish, qayta nomlash, joyni o'zgartirish va o'chirish buyruqlarini qulaykok bajaradi;

Turli matnli fayllar, ma'lumotlar bazasi matnlarini kurish, matnli fayllarni tahrirlash imkoniga ega.

Utilitalar va avtonom dasturlar tor ixtisoslashgan bo'lib, har biri o'z vazifasini bajaradi. Biroq utilitalar avtonom dasturlardan farqli ravishda tegishli qobiqlar muhitida bajaradi. Qobiq foydalanuvchiga sifat jihatdan yangi interfeys taqdim etadi. OT foydalanuvchi operatsiya va buyruqlarini ikir-chikirigacha bilishdan ozod etadi. Utilitalar foydalanuvchiga qo'shimcha xizmatlarni asosan disklar va faylli tizimlar bo'yicha xizmat ko'rsatish ko'rinishida taqdim etadi.

Utilitalar quyidagi vazifalarni bajarishga yo'l qo'yadi:

- diskklarga xizmat ko'rsatish;
- fayl va kataloglarga xizmat ko'rsatish (xuddi qobiqlar kabi);
- arxivni yaratish va yangilash;
- turli rejim va formatlarda matnli va boshqa fayllarni bosish;
- kompyuterni virusdan himoya qilish.

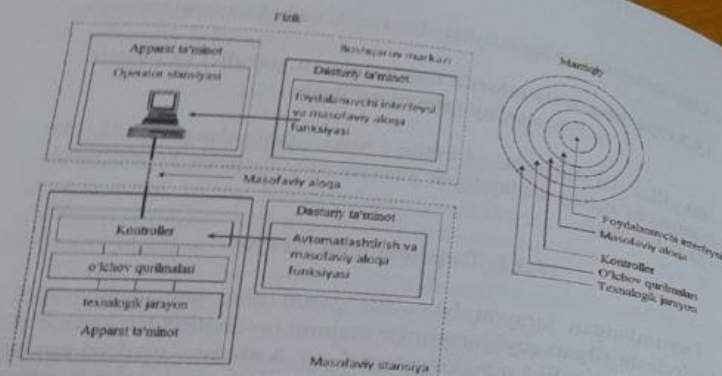
Operatsion tizimning ixtiyeriy buyrugini va hokazolarni bajaradi.

15.3. Operatsion tizimlarda taqsimlangan, parallel va bulutli jarayonlarni qo'llash

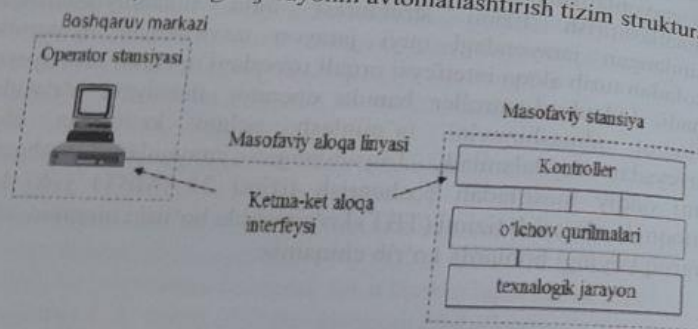
80-yillar o'rtalarida, tarmoq yoki taqsimlangan OT lar boshqaruvchi ostida ishlaydigan shaxsiy kompyuterlar keskin tarzda rivejlana boshladi.

15.3.1. Taqsimlangan jarayon

Taqsimlangan jarayon bu – bir guruh qilib jamlangan o'zaro mahalliy lashtirilgan quyi jarayonlar majmui bo'lib nisbatan katta fizik (yoki hatto geografik) maydongatarqalgan, boshqaruv markazi esa bu quyi jarayonlardan ancha uzoqda bo'lishi mumkin. 15.1.-rasmda ushbu avtomatlashtirish tizimi strukturasi bitta mahalliy lashtirilgan taqsimlangan jarayondagi quyi jarayon tasvirlangan. Kontroller masofadan turib aloqa interfeysi orqali uzoqdagi operator stansiyasiga ulanadi. Odatda kontroller hamda operator stansiyasi o'rtasidagi ma'lumot almashinuvini ta'minlash uchun ketma-ket aloqa interfeysdan foydalaniladi. Taqsimlangan jarayonlarni boshqarish tizimi oddiy masofadan boshqarish tizimi 24 (MBT) yoki katta tarmoqni boshqarish tizimi (TBT) ko'rinishida bo'lishi mumkin. ularni kengroq keyingi boblarda ko'rib chiqamiz.



15.1. Taqsimlangan jarayonni avtomatlashtirish tizim strukturasi.



15.2. Masofadan boshqarish tizimi

15.3.2. Parallel jarayonlar

Jarayonning o'zaro ta'siri parallel jarayonlarning bir-biri bilan aloqa qilish mexanizmlari bilan bog'liq. O'zaro aloqalarning eng keng tarqalgan shakllari umumiy xotira va xabarlarni uzatishdir, ammo o'zaro ta'sir yashirin bo'lishi mumkin (dasturchiga ko'rinmaydi).

Umumiy xotira - bu jarayonlar o'rtasida ma'lumotlarni uzatishning samarali vositasi. Umumiy xotira modelida parallel jarayonlar asinxron tarzda o'qigan va yozgan global manzil maydonini bo'lishadi.

Asenkron bir vaqtda kirish olib kelishi mumkin poyga shartlari kabi mexanizmlar qulflar, semaforalar va monitorlar bulardan saqlanish ko'plab parallel dasturlash tillari va kutubxonalari kabi umumiy xotirani bevosita qo'llab-quvvatlaydi Cilk, OpenMP va Qurilish bloklarini burish, ekspluatatsiya qilish uchun mo'ljallangan.

Xabar uzatish modelida parallel jarayonlar xabarlarni bir-birlariga uzatish orqali ma'lumotlarni almashadilar. Ushbu aloqalar asenkron bo'lishi mumkin, bu erda qabul qiluvchi tayyor bo'lguncha xabar yuborilishi mumkin yoki qabul qiluvchi tayyor bo'lishi kerak bo'lgan sinxron. Ketma-ket jarayonlarni etkazish (CSP) xabarlarni rasmiylashtirish jarayonlarni ulash uchun sinxron aloqa kanallaridan foydalanadi va Okkam, Limbo va Boring kabi muhim tillarga olib keldi. Aksincha, aktyor modeli asenkron xabar uzatishni ishlatadi va kabi tillarni loyihalashda ishlatilgan

Massivli-parallel kompyuterlar paydo bo'lishi bilan parallel jarayonlarni aloqasini qo'llab-quvvatlovchi kutubxona interfeyslar keng tarqaldi. Bu yo'nalishning toifali vakiliga Message Passing Interface (MPI) interfeysi misol bo'ladi. Bu interfeys amalda vektor-konveyerli super - EXM dan tortib shaxsiy kompyutergacha bo'lgan barcha parallel platformalarda mavjud. Dasturning qaysi parallel jarayonlari dasturning qaysi qismida va jarayonlar bilan ma'lumotlar almashishi yoki o'z ishini sinxronlab borishi kerakligini dasturchining o'zi belgilaydi. Odatda parallel jarayonlarning manzil soxasi turlicha bo'ladi. Xususan, bu g'oyaga MPI va PVI da amal qilinadi. Boshqa texnologiyalarda, masalan Shmem da lokal (private) va umumiy (shared) o'zgaruvchilari qo'llaniladi. Bu o'zgaruvchilarga dasturning barcha jarayonlari murojat etishi mumkin va PutG'Get toifasidagi operatsiyalar yordamida umumiy xotira bilan ishlash usuli tashkil etiladi. Linda sistemasi o'ziga xos xususiyatga ega bo'lib, unda ixtiyoriy ketma-ketlikda tilga to'rtta: in, out, read va eval funktsiyalarini qo'shadi va parallel dasturlar tuzish imkonini beradi. Afsuski, ushbu keltirilgan g'oya oddiy bo'lishiga qaramasdan, uni amalda qo'llash muammolar tug'diradi.

Taqsimlangan xotira parallel kompyuterlardagi keng tarqalgan dasturlash texnologiyasi MPI texnologiyasi xisoblanadi. Bunda sistemalardagi parallel jarayonlarning o'zaro muloqat usuli bir-bi-

bilan xabarlar almashishdan iborat. Bu usul texnologiyasi nomi - Message Passing Interface (xabarlar almashish interfeysi) deb aks ettirilgan. MPI standartida sistema amal qilishi va dastur yaratishda foydalanuvchilar amal qilishi kerak bo'lgan qoidalar mavjud. MPI Fortran va Si bilan ishlashni qo'llab-quvvatlaydi. Interfeysning to'liq versiyasida 125 tadan ko'proq protsedura va funktsiyalar mavjud. MPI MIMD (Multiple Instruction Multiple Data) stilidagi parallel dasturlarni qo'llab quvvatlaydi. Unda turli matnlar bilan berilgan dasturlarni birlashtiriladi. Biroq bunday dasturlarni tuzish va otladka etish murakkab. Shuning uchun amalda dasturchilar SPMD (SINGLE PROGRAM MULTIPLE DATA) parallel dasturlash modelidan foydalanishadi. Unda parallel jarayonlar uchun bitta dastur kodini qo'llaniladi. Xozirgi paytda MPI ko'proq to'rlar bilan ishlashni qo'llab-quvvatlaydi. MPI kutubxonasining parallel jarayonlar bilan mo'ljallangan funktsiyalarini qo'llab-quvvatlash uchun dasturni kompilyatsiya etishda zarur kutubxona modullarini ulash zarur. Buni buyruq sifatida ulash yoki ko'pgina sistemalarda maxsus mpicc (Si tilidagi dasturlar uchun), mpicc (SiQQ tilidagi dasturlar uchun) va mpif77/mpif90 (Fortran77/90 tilidagi dasturlar uchun) buyruq va funktsiyalar mavjud. Kompilyatorning "-o name" bo'limi yaratiladigan ishchi fayl a.out deb nomlanadi, Masalan: mpif77-o program program.f

Ishchi fayl yaratilgandan keyin uni zarur miqdordagi jarayonlar uchun ishga tushirish kerak. Buning uchun odatda MPI-dasturlarini ishga tushiruvchi mpirun buyrug'i mavjud, Masalan: mpirun - np N< dastur va argumentlari>

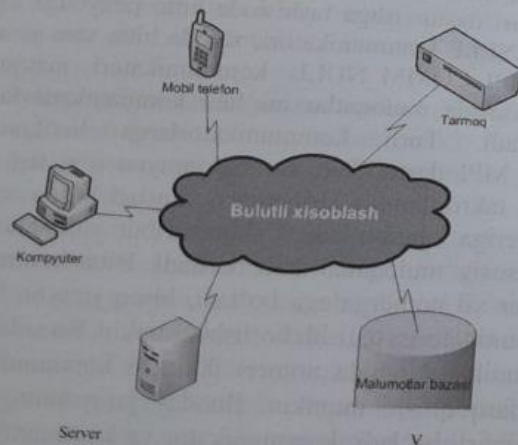
Bu erda N - bitta masaladagi jarayonlar miqdori. Ishga tushirilgandan keyin bitta dastur ishga tushirilgan jarayonlar tomonidan chaqiriladi. Chaqirish natijasi sistemaga bog'liq ravishda terminalga yoki nomi ko'rsatilgan faylga yoziladi. Qolgan barcha ob'ektlar: protseduralar, konstantalar va MPI da aniqlangan ma'lumotlar toifalari MPI old qo'shimchaga ega bo'ladi. Agar foydalanuvchi dasturda bunday qo'shimchali nomlardan foydalanmasa xam MPI ob'ektlar bilan muammo tug'ilmaydi. Bundan tashqari Si tilida funktsiya nomlaridan bosh va kichik xarflar farqlanadi. Odatda MPI funktsiya nomidagi MPI qo'shimchasidan keyingi xarf bosh xarf bilan, keyingilari kichik xarflar bilan yoziladi. MPI konstantalar nomi esa

butunligicha bosh xarflar bilan yoziladi. MPI interfeysining tavsifi mpif.h (mpi.h) faylida mujassamlashtirgan. Shuning uchun MPI - dastur boshqa include 'mpif.h' direktivasi (ko'rsatmasi) (Si tilida muloqat qiluvchi parallel dasturlar to'plamidir. Har bir jarayon bir marta yuz beradi va dasturni parallel qismini tashkil etadi. MPI dastur bajarilishi jarayonida qo'shimcha jarayonlarni tashkil etish yoki mavjudlarini yo'qotish mumkin emas. Har jarayon o'zining manzil soxasida joylashadi va umumiy o'zgaruvchilar MPI da yo'q. Jarayonlar o'rtasida muloqatning yagona usuli xabarlar almashishdir. Parallel jarayonlarni muloqatini mustaqil amalga oshirish uchun jarayonlar guruxi tanlab olinib ular uchun aloxida muxit kommunikatorlar yaratiladi. Jarayonlar faqat bitta kommunikatorlar ichida muloqat qililadi va turli kommunikatorlarga yuborilgan xabarlar kesishmaydi. Fortran tilidagi kommunikatorlar MPI_COMM_WORLD toifasiga ega bo'ladi. Dastur ishga tushishida yuzaga keladigan jarayonlar to'la qamrovli MPI_COMM_WORLD nomli kommunikator doirasida ishlaydi deb xisoblanadi. Ushbu kommunikator doimo mavjud bo'ladi va ishga tushgan barcha MPI dasturlarni muloqat qilish uchun xizmat qiladi. Bundan tashqari dastur ishga tushishida bitta jarayonga ega bo'lgan MPI_COMM_SELF kommunikatori, xamda bitta xam jarayonga ega Jarayonlar o'rtasida muloqatlar ma'lum kommunikatorlar doirasida amalga oshadi. Turli kommunikatorlarga berilgan xabarlar kesishmaydi. MPI dasturidagi xar bir jarayon o'zi tegishli bo'lgan gurux ichida takrorlanmas atributga - musbat butun sondan iborat jarayon nomeriga ega bo'ladi. Ushbu atribut yordamida jarayonlar o'rtasidagi asosiy muloqatlar olib boriladi. Bitta kommunikatoridagi jarayonlar xar xil nomerga ega bo'ladi, biroq jarayon bir vaqtda bir necha kommunikatorga tegishli bo'lishi mumkin. Bu xolda jarayonning bitta kommunikator ichida nomeri ikkinchi kommunikator ichidagi nomeridan farq qilishi mumkin. Bunday jarayonning ikkita asosiy atributi tushinarli bo'ladi: kommunikator va kommunikator nomi. Agar guruxda n ta jarayon bo'lsa, u xolda joriy guruxdagi ixtiyoriy jarayon nomeri 0 dan n - 1 gacha oraliqda yotadi. Jarayonlarni o'zaro muloqat qilishining asosiy usuli xabarlar almashishdan iborat. Xabar - bu biror toifadagi ma'lumotlar to'plamidir. Xar bir xabar bir qancha

atributlarga ega bo'ladi. Masalan, jo'natuvchi jarayon nomeri, qabul qiluvchi jarayon nomeri, xabar identifikatori va boshqalar. Xabarning asosiy atributlaridan biri uning identifikatori yoki tegi hisoblanadi. Xabarni qabul qilayotgan jarayon xabar identifikatori bo'yicha unga bitta jarayondan kelgan ikkita xabarni farqlab oladi. Xabar identifikatori musbat butun son bo'lib 0 va MPI_TAG_UP diapazonida yotadi va MPI_TAG_UP 32767 dan kichik emas. Xabar atributlari bilan ishlash uchun massiv (Si tilida struktura) kiritilgan. Massiv elementlari yordamida xabar qiymatlariga murojat etiladi. MPI protseduralarini ko'pchiligida oxirgi argumentda muvaffaqiyatli tugaganlik xaqidagi ma'lumot qaytariladi. Muvaffaqiyatli bajarilganda MPI_SUCCES qiymati, aks xolda xatolik kodi qaytariladi. Protsejura bajarilishida yuz bergan xatolikni uning tavsifidan bilib olishi mumkin. Turli xatolik kodlariga mos tavsiflar mpif.h faylida joylashadi.

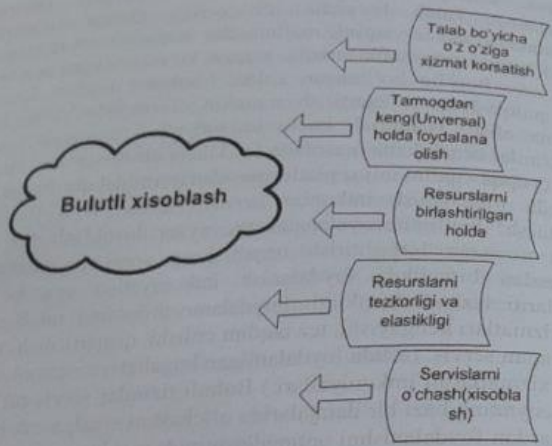
15.3.3. Bulutli jarayonlar

Bulutli texnologiyalar - bu model iste'molchiga ATni servis sifatida internet orqali namoyon qiladi.



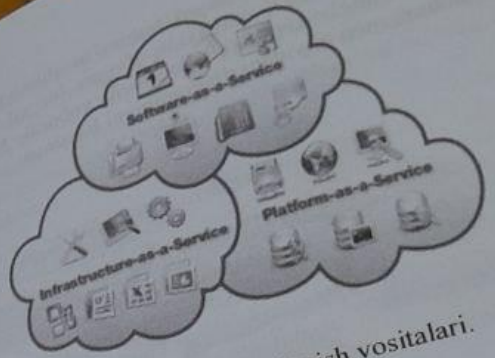
15.3-rasm. Bulutli texnologiyani asosiy model tavsiflari

Ularni, boshqa turdagi hisoblashlardan farqlash (internet resurslaridan). Talab bo'yicha o'z o'ziga xizmat ko'rsatish. Foydalanuvchi server vaqтини, ma'lumotlar saqlash ombori xajmini, zarur bo'lganda avtomatik tarzda, xizmat ko'rsatayotgan provayder bilan o'zaro bog'liq bo'lmagan xolda, hisoblash kuchini mustaqil tarzda aniqlash va o'zgartirish mumkin. Tarmoqdan standart foydalana olish. Hisoblash kuchi imkoniyatlari tarmoqda standart mexanizimlar orqali katta masofada foydalana olish mumkin. Har - xil turdagi (yupqa - qalin) mijoz platformasidan (terminal qurilmalar) keng qamrovda foydalanish imkonini beradi. Server V 8 Resurslarni birlashtirish. konfiguratsiyalangan provayder hisoblash resurslarini Resurslarni tezkor elastikligi. Foydalanuvchilarning talabiga qarab bulut xizmatlari kengayishi, tez taqdim etilishi, qisqartirilishi mumkin. O'Ichangan servis. (aslida foydalanilgan bugalteriya istemol servisi va to'lov xizmatlarini imkoniyatlari.) Bulutli tizimlar turiga qarab abstraksiyaning bazi bir darajalarida o'Ichashni amalga oshirish orqali resurslardan foydalanishni optimallashtiradi va ular ustidan avtomatik nazorat qiladi.



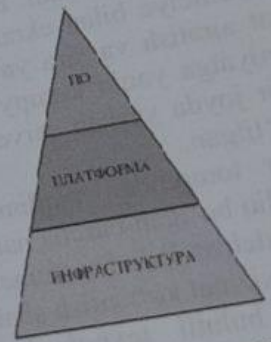
15.4-rasm. Asosiy xarakteristik modellar

Aslida, faqat farq faqat ma'lumotlarni saqlash va qayta ishlash usuli yotadi. Barcha operatsiyalar (uning kuchi yordamida) kompyuteringizga sodir bo'lsa, u - bo'lmagan bir "bulut", va jarayon tarmoq ustida serverda shakllangan bo'lsa, bu tendentsiya narsa, va bu deb ataladi "cloud computing". Boshqa so'zlar bilan aytganda, bulutli hisoblash - ularning maqsadlari, vazifalari va loyihalarni erishish uchun apparat, dasturiy ta'minot, metodologiyasi va Internet xizmatlari kabi foydalanuvchiga mavjud vositalar turli hisoblanadi.



15.5-rasm - Foydalanish vositalari.

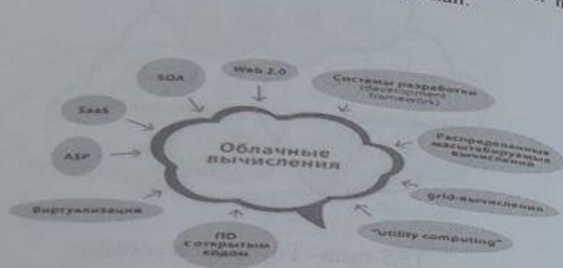
Amaliyot shuni ko'rsatadiki, "bulutli texnologiyalar" / "bulutli xizmat" tushunchalari, "bulutlar" shaklida, umuman qabul qilingan grafik tasviri bilan 12 foydalanuvchilarni shunchaki chalkashtirib yuboradi, aslida ularning tuzilishi keyingi piramida shaklida ifodalanadi.



15.6-rasm Xizmatlar taqdim etish infrastruktura satxlari.

Piramidaning "infratuzilmasi" - bu jismoniy qurilmalar majmuasi (serverlar, qattiq disklar, va boshqalar). Buning ustiga, "platform" foydalanuvchilarning iltimosiga ko'ra xizmatlarning to'plami va yuqori

dasturiy ta'minot. Bundan tashqari, bulutni hisoblash texnologiyasi va yondashuvlarining sintezi natijasida olingan bir xil bazaviy vektor ekanligini bilishingiz kerak (chalkashtirib yubordi. Nimani nazarda tutishim uchun, men quyidagi diagrammani beraman:



15.7-rasm Diagrammasi.

Endi u bir oz ko'proq ayon bo'ldi, foyda sxemasi juda oddiy, deb o'ylayman. Biroq, umuman, bulut - bu kompyuteringizni resurslarni bevosita ishtirokisiz hisoblash serverlar va boshqa dona bajaradi pyuresi bunday turdagi hisoblanadi. Balki, biz barcha shuning aslida, faqat bitta mikroishlemciye bilan ekranning bo'ladi va barcha hisob-kitoblar va quvvat ajratish va sira ya'ni amalga oshiriladi, birinchi, gapirish va hokimiyatga yaqin kompyuterlar, qaytib borib, shunday qilib, sodir bor, bir joyda yashab serverlarida, masalan, bulut ichida qayta-qayta aytib o'tilgan.

Bulutli tizimlar tomonidan taqdim etilgan xizmatlar: -Cloud hisoblash bilan bog'liq bo'lgan barcha narsalar (bundan keyin CC deb ataladi) odatda AAA deb ataladi. Bu shunchaki shifrlangan - "xizmat", ya'ni "xizmat" yoki "xizmat ko'rsatish shaklida".

Bugungi kunda bulutli texnologiyalar va, aslida, ularning kontseptsiyasi o'z foydalanuvchilariga quyidagi turdagi xizmatlarni taqdim etishni o'z ichiga oladi: - Xizmat-as-saqlash ("xizmat sifatida saqlash"). Bu ehtimol SS-xizmatlaridan eng oson, ya'ni diskdagi bo'sh joy. Bizning har birimiz monitorda dahshatli ogohlantirish paydo bo'lgan vaziyatga duch keldik: "Mantiqiy disk bo'sh joyni bo'shatish,

keraksiz dasturlarni yoki ma'lumotlarni o'chirish uchun to'la". Saqlash as-a-Service xizmati tashqi xotiradagi ma'lumotlarni, "bulut" da saqlash imkonini beradi. Siz uchun bu qo'shimcha mantiqiy disk yoki papka kabi ko'rinadi. Xizmat qolganlari uchun asosdir, chunki u deyarli har birining bir qismidir. Masalan, Google Drive va boshqa shunga o'xshash xizmatlar.

15.4. Nazorat savollari

1. Operatsion tizim va tarmoqlarning istiqboli
2. IBM firmasining operatsion tizimlari.
3. HP firmasining operatsion tizimlari.
4. OracleG'Sun va boshqa firmalarning Operatsion tizimlari.
5. Operatsion qobiqlar haqida ma'lumot
6. Operatsion tizimlarda taqsimlangan, parallel va bulutli jarayonlarni qo'llash.
7. Taqsimlangan jarayon
8. Xabarlarni almashish interfeys texnologiyasi
9. Bulutli jarayonlar

ADABIYOTLAR RO'YXATI

1. Е. Таненбаум, Х. Бос. Современные операционные системы. 4-е изд. – СПб.: Питер, 2015. 1120 с.
2. А.В. Замятин. Операционные системы. Теория и практика: учебное пособие – Томск: Изд-во Томского политехнического университета, 2012. 263 с.
3. С.В. Сеницын, А.В. Батаев, Н.Ю.Налютин. Операционные системы: учебник для студ. Учреждений высш. проф. образования. 3-е изд., стер – М. : Издательский центр «Академия», 2013. – 304 с.
4. Andrew S. Tanenbaum, Херберт Бос. Модерн. Оператинг Системс, 4th Едитион Пеарсон. УК Лондон. 2014.
5. Гордеев А.В.«Операционные системы» Учебное пособие, Москва: Изд.дом: «Питер», 2008, - 384 стр.
6. Олифер В.Г., Олифер Н.А., «Сетевые операционные системы», М:Питер, 2006, 560 стр.
7. Andrew S. Tanenbaum, Albert S. Woodhull. Operating Systems: Design and Implementation, Third Edition. Prentice Hall. 2006 – 1080p.
8. Arash H. L., Mohammadreza M. Mobile Operating Systems and Programming. Andrew S. Tanenbaum. Distributed operating systems.
9. Таненбаум Е. Архитектура компьютера. 5 – е изд. – СПб.: Питер, 2007.– 844с.
10. Trent Jaeger. Operating Systems Security.
11. Sibsankar Haldar. Alex Alagarsamy Aravind. Operating Systems. 2010 -563p.
12. Craig Hunt, Robert Bruce Thompson. Windows NT TCP/IP Network Administration. O'Reilly Media. 1998. – 512p.
13. Иртегов Д.В. Введение в операционных системы. 2 издание. БХВ–Петербург. 2008 – 1040 стр.
14. Кондратов В.К. Введение в операционную систему УНИХ. 2002.-93 стр.
15. Курячий Г. Маслинский К. Операционная система Линух. Учебное пособие.

MUNDARIJA

Kirish.....	3
I. OPERATSION TIZIMLAR FANINING MAZMUNI VA MOHIYATI.....	5
1.1. Operatsion tizim tushunchasi.....	5
1.2. Kompyuter sinflarining sinflanishi.....	6
1.3. Hisoblash tizimlarining tarkibiy qismlari.....	7
1.4. Operatsion tizim kengaytirilgan mashina va resurslarni boshqaruvchi sifatida.....	12
1.4.1. Operatsion tizim kengaytirilgan mashina sifatida.....	12
1.4.2. Operatsion tizim resurslarni boshqaruvchi sifatida.....	13
1.5. Nazorat savollari.....	14
II. OPERATSION TIZIM QURISH TAMOYILLARI.....	16
2.1. Operatsion tizimni qurish asosiy printsiplari.....	16
2.1.1. Chastota printsiipi.....	16
2.1.2. Modullilik printsiipi.....	17
2.1.3. Funktsional tanlanish printsiipi.....	17
2.1.4. Operatsion tizimni generatsiya qilish printsiipi.....	17
2.1.5. Funktsional ortiqchalilik printsiipi.....	18
2.1.6. Standart holatlar printsiipi (po umolchaniyu).....	18
2.1.7. Joyini o'zgartirish printsiipi.....	18
2.1.8. Virtuallashtirish printsiipi.....	18
2.1.9. Dasturiy ta'minotni Tashqi qurilmalarga bog'liq emasligi (muustahilligi) printsiipi.....	20
2.1.10. Mutanosiblik printsiipi (sovmestimost).....	20
2.1.11. Ochiqlik va qo'shimcha imkoniyatlar qo'shish printsiipi.....	21
2.1.12. Mobillilik printsiipi (ko'chirib o'tkazish).....	21
2.1.13. Xavfsizlik printsiipi.....	22
2.2. Nazorat savollari.....	23
III. OPERATSION TIZIMLARNING ARXITEKTURASI.....	24
3.1. Operatsioon tizim arxitekturasi (Yadro, buyruqlar protsessori, kiritish – chiqarish tizimi, fayl tizimi).....	24
3.2. Ko'p protsessorli ishlov berish, paketli ishlov berish, real vaqt tizimlari, vaqt ajratish, taqsimlangan tizimlar.....	
3.3. Ob'ektga yo'naltirilgan yondashish.....	

3.4. Klent server modellari.....	31
3.5. Nazorat savollari.....	35
IV. OPERATSION TIZIMLARDA RESURS VA JARAYON TUSHUNCHASI.....	37
4.1. Operatsion tizimlarda jarayon va resurs tushunchasi.....	37
4.1.1. Operatsion tizimda jarayon tushunchasi.....	37
4.1.2. Operatsion tizimda resurs tushunchasi.....	37
4.2. Resurslarni umumiy sinflarga ajratilishi.....	38
4.3. Jarayonlar xolati.....	38
4.3.1. Jarayon konteksti va Process Control Block (jarayon diskriptori).....	43
4.3.2. Bir martalik amallar (operatsiyalar).....	46
4.3.3. Ko'p martalik operatsiyalar.....	46
4.4. Jarayonlarni rejalashtirish algoritmlari va parametrlari.....	48
4.4.1. Rejalashtirish ko'rsatkichi va algoritmlarga talablar.....	49
4.4.2. Rejalashtirish parametrlari.....	49
4.5. Semaforalar.....	51
4.6. Nazorat savollari.....	53
V. OPERATSION TIZIMLARDA JARAYONLARNI BOSHQARISH.....	55
5.1. Operatsion tizimda jarayonlarni boshqarish.....	56
5.2. Rejalashtirish, dispetcherlash va sinxronizatsiyalash.....	57
5.2.1. Jarayon va oqimlarni rejalashtirish va dispetcherlash.....	57
5.2.2. Sinxronlashtirish muammolari.....	58
5.3. Nazorat savollari.....	58
VI. OPERATSION TIZIMLARDA OQIM(THREADS)LAR.....	59
6.1. Operatsion tizimlarda oqim(threads) tushunchasi.....	59
6.2. Darsturlarning ko'p oqimli (Multithreading) bajarilishi.....	60
6.3. Nazorat savollari.....	62
VII. OPERATSION TIZIMLARDA TUPIK (DEADLOCK) TUSHUNCHASI.....	63
7.1. Tupik muammolari.....	63
7.2. Resurslar taqsimlash grafi.....	64
7.3. Bo'shatiladigan va bo'shatilmaydigan resurslar.....	64
7.4. O'zaro berk hodisalar.....	66
7.5. O'zaro berk hodisalarni modellashtirish.....	67
7.6. O'zaro berk hodisadan chiqish.....	68

7.7. Tupiklarni qayta ishlash usullari.....	69
7.7.1. Resursni ustuvor o'zlashtirish hisobiga tiklash.....	69
7.7.2. Orqaga qaytarish orqali tiklash.....	69
7.7.3. Jarayonlarni yo'q qilish orqali tiklash.....	70
7.8. Tupiklarni oldini olish.....	71
7.9. Nazorat savollari.....	72
VIII. OPERATSION TIZIMDA XOTIRANI BOSHQARISH.....	73
8.1. Xotira va uning aksi, virtual address makoni.....	73
8.2. Xotirani boshqarishning umumiy tamoyillari.....	77
8.2.1. Xotira boshqaruvining eng oddiy sxemalari.....	77
8.2.2. Qat'iy belgilangan(fiksirlangan) bo'limli sxemalar.....	78
8.3. Xotirani statik va dinamik taqsimlash, segmentli, sahifali, segment – sahifali tashkil etish.....	79
8.3.1. Dinamik taqsimlanish. Almashtirish(svoping).....	82
8.3.2. O'zgaruvchan bo'limli sxemalar.....	82
8.3.3. Saxifali xotira.....	83
8.3.4. Segmentli va segment – sahifali xotira.....	84
8.4. Nazorat savollari.....	85
IX. VIRTUAL XOTIRA.....	86
9.1. Virtual xotira konsepsiyasi.....	87
9.2. Virtual xotirani saxifali tashkil etish.....	90
9.3. FIFO, LRU va "ikkinchi imkoniyat" algoritmlari.....	94
9.4. Nazorat savollari.....	98
X. OPERATSION TIZIMLARDA FAYL TIZIMLARI.....	99
10.1. Fayl tizimi funksiyalari va ma'lumotlar ierarxiyasi.....	99
10.2. Fayllarni joylashtirish jadvali.....	101
10.3. Fayl tizimi imkoniyatlari va ishonchliligi.....	102
10.4. FAT, NTF, HPES, ext2, ext3, ext4 va boshqa fayl tizimlari.....	103
10.4.1. FAT, VFAT va FAT32 fayl tizimlari.....	103
10.4.2. HPFS fayl tizimi.....	10
10.4.3. NTFS fayl tizimi.....	106
10.4.4. ext2, ext3, ext4 fayl tizimlari.....	1
10.5. Nazorat savollari.....	
XI. VIRTUAL FAYL TIZIMLARI (VFS) VA TARMOQ FAYL TIZIMLARI (NFS).....	106
11.1. Tashqi xotirani boshqarish.....	

11.2. Keshlash.....	118
11.3. Tranzaksiya asosida fayl tizimlar.....	120
11.4. Tarmoq fayl tizimlar(NTF).....	121
11.5. Nazorat savollari.....	130
XII. MA'LUMOTLARNI KIRITISH – CHIQARISH TIZIMI.....	131
12.1. Kompyuter tizimida ma'lumotlarni kiritish-chiqarish tizimini tashkil etish.....	131
12.2. Kontrollerlar.....	132
12.3. Drayverlar.....	133
12.4. Nazorat savollari.....	135
XIII. OPERATSION TIZIM VA TARMOQLARNING XAVFSIZLIGI.....	136
13.1. Xavfsizlik kontseptsiyasi.....	136
13.2. Tarmoqli va tizimli tahdidlar(xujumlar).....	137
13.3. Tarmoq tizimlarining auditi.....	139
13.4. Kompyuterlarning xavfsizlik darajalari.....	141
13.5. Nazorat savollari.....	143
XIV. BULUTLI HISOBLASH (CLOUD COMPUTING) UCHUN OPERATSION TIZIMLAR.....	144
14.1. Cloud computing tushunchasi.....	144
14.2. Xizmatlar, ma'lumotlarni qayta ishlash markazlari.....	146
14.2.1. Xizmat sifatida infratuzilma (IaaS).....	146
14.2.2. Xizmat sifatida platforma (PaaS).....	146
14.2.3. Dastur xizmat sifatida (SaaS).....	147
14.3. Cloud computing uchun operatsion tizimlar dasturiy vositalar obzori.....	147
14.3.1. Bulutli dastur.....	148
14.3.2. Ta'lim uchun Microsoft buluti.....	149
14.4. Nazorat savollari.....	150
XV. OPERATSION TIZIM VA TARMOQLARNING ISTIQBOLI.....	151
15.1. IBM, HP, OracleG'Sun va boshqa firmalarning Operatsion tizimlari.....	151
15.2. Operatsion tizimlarning qobiqlari.....	156
15.3. Operatsion tizimlarda taqsimlangan, parallel va bulutli jarayonlarni qo'llash.....	157
15.3.1. Taqsimlangan jarayon.....	157

15.3.2. Xabarlarni almashish interfeys texnologiyasi.....	158
15.3.3. Bulutli jarayonlar.....	162
15.4. Nazorat savollari.....	167
ADABIYOTLAR RO'YXATI.....	168

ABDULLAYEVA G.X., HAMRAYEVA S.I.,
XUDAYBERGANOV T.R., XUDAYBERGANOV O.F.

OPERATSION TIZIMLAR

O'QUV QO'LLANMA

Toshkent - "METHODIST NASHRIYOTI" - 2024

Muharrir: Bakirov Nurmuhammad

Texnik muharrir: Tashatov Farrux
Musahhih: Hazratqulova Ruxshona
Dizayner: Ochilova Zarnigor

Bosishga 1.04.2024 da ruxsat etildi.
Bichimi 60x90. "Times New Roman" garniturası.
Ofset bosma usulida bosildi.
Shartli bosma tabog'i 11. Nashr bosma tabog'i 11.
Adadi 300 nusxa.

"METHODIST NASHRIYOTI" MCHJ matbaa bo'limida chop etildi.
Manzil: Toshkent shahri, Shota Rustaveli 2-vagon tor ko'chasi, 1-uy.



+99893 552-11-21

Nashriyot roziligrisiz chop etish ta'qilnadi.

ISBN 978-9910-03-210-3



9 789910 032103

