

MASHINALI DARAJADAGI DASTURLAR: OPERATSION TIZIMLAR VA DRAYVERLARDA QO‘LLANILISHI.

Umarov Begzodbek Azizovich

Farg‘ona davlat universiteti amaliy matematika kafedrası o‘qituvchisi

ubaumarov@mail.ru

Iminova Gavharoy Ilhomjon qizi

Farg‘ona davlat universiteti talabasi

gavharoyiminova190@gmail.com

Annotatsiya: Ushbu maqolada mashinali darajadagi dasturlarning operatsion tizimlar va drayverlar sohasida ahamiyati, ularning tizim ishlashiga ta’siri va zamonaviy kompyuter arxitekturasida sharoitida qo‘llanilishi tahlil qilinadi. Mashinali darajadagi kodlar apparat darajasida tizimni boshqarish imkonini beradi, shuning uchun operatsion tizim yadrosi va drayverlar ishlab chiqishda keng qo‘llaniladi. Maqolada mashinali darajadagi dasturlarning struktura va funksiyalariga e’tibor qaratilib, ularning samaradorligi misollar orqali ko‘rsatib o‘tiladi.

Kalit so‘zlar: Mashinali darajadagi dasturlar, operatsion tizim, drayver, kernel, apparat bilan muloqot, protsessor buyruqlari, tizim boshqaruvi, samaradorlik, interrupt, optimallashtirish.

Abstract: This article analyzes the importance of machine-level programs in the fields of operating systems and drivers, their impact on system performance, and their application in modern computer architecture. Machine-level codes enable system control at the hardware level, which is why they are widely used in the development of operating system kernels and drivers. The article focuses on the structure and functions of machine-level programs, demonstrating their efficiency through examples.

Keywords: Machine-level programs, operating system, driver, kernel, hardware interaction, processor instructions, system management, efficiency, interrupt, optimization.

Аннотация: В данной статье анализируется значение программ машинного уровня в области операционных систем и драйверов, их влияние на работу системы

и применение в современной компьютерной архитектуре. Машинные коды обеспечивают управление системой на уровне аппаратного обеспечения, поэтому широко используются при разработке ядра операционной системы и драйверов. В статье уделяется внимание структуре и функциям программ машинного уровня, их эффективности, продемонстрированной на примерах.

Ключевые слова: Программы машинного уровня, операционная система, драйвер, ядро, взаимодействие с аппаратным обеспечением, инструкции процессора, управление системой, эффективность, прерывание, оптимизация.

Kirish

Zamonaviy kompyuterlar murakkab tizimlardan tashkil topgan bo‘lib, ular apparat va dasturiy ta’minotning bir-biri bilan uzviy bog‘liqligi orqali samarali ishlaydi. Ushbu tizimlarning asosi - operatsion tizim va qurilma drayverlari bo‘lib, ular apparat resurslarini boshqarish va foydalanuvchi dasturlariga xizmat ko‘rsatish vazifasini bajaradi. Operatsion tizim va drayverlarning pastki qatlamlari, ya’ni yadro (kernel) va qurilma boshqaruv kodi, mashinali darajadagi dasturlardan iborat bo‘lishi zarur. Chunki yuqori darajadagi dasturlash tillari apparatga to‘g‘ridan-to‘g‘ri kira olmaydi, ular bir qatlam orqada joylashgan. Mashinali darajadagi dasturlar esa protsessor buyruqlarining to‘liq to‘plami bo‘lib, apparatni nazorat qilish va tizim ish faoliyatini boshqarishda eng samarali vositadir. Operatsion tizim yadrosi vazifalarining murakkabligi va tezkorligi, shuningdek, drayverlarning qurilmalar bilan o‘zaro ta’sirda kechikishlarni minimallashtirish talablarining yuqoriligi mashinali darajadagi kodlardan keng foydalanishga asos bo‘ladi. Shu bilan birga, mashinali darajadagi dasturlar tizimning xavfsizligi va ishonchliligini ta’minlashda ham muhim rol o‘ynaydi.

Asosiy qism

Mashinali darajadagi dasturlar protsessor tomonidan to‘g‘ridan-to‘g‘ri bajariladigan ko‘rsatmalar to‘plamidir. Bu dasturlar yuqori darajadagi tillardan farqli ravishda, hech qanday qo‘shimcha tarjimon (kompilyator yoki interpretator) talabi bilan ishlaymaydi. Har bir mashinali buyruq protsessor arxitekturasiga muvofiq yozilgan bo‘lib, ularning ketma-ketligi hisobga olinadi. Mashinali kodning afzalliklari quyidagilar: Kodek (ya’ni dastur kodi) to‘g‘ridan-to‘g‘ri protsessor tomonidan bajariladi, bu maksimal ishlash tezligini

ta'minlaydi. Xotira va protsessor vaqtining tejashiga imkon beradi. Registrlar, portlar, xotira manzillari kabi apparat komponentlariga kirish imkoniyati. Kamchiliklari esa: Kod yozish va o'qish qiyin, dasturchidan yuqori malaka talab qiladi. Har bir protsessor arxitekturasi uchun kod alohida yozilishi zarur. Texnik qo'llab-quvvatlash qiyinligi. Operatsion tizim yadrosi - tizimning eng muhim va markaziy qismi hisoblanadi. U kompyuter resurslarini boshqaradi, jarayonlarni muvofiqlashtiradi, xotirani ajratadi va apparat qurilmalari bilan muloqot qiladi. Ushbu funktsiyalarni tez va ishonchli bajarish uchun yadroning bir qismi mashinali darajadagi dasturlar orqali amalga oshiriladi. Mashinali kodning yadrodagi asosiy roli: Jarayonlar orasidagi almashtirish, ustuvorlik belgilash va vaqt bo'laklarini taqsimlash (scheduling) tezkor bajarilishi kerak. Tashqi qurilmalardan kelgan signal va voqealarga darhol javob qaytarish. Fizik va virtual xotira manzillarini nazorat qilish. Kiritish/chiqarish portlari va registrlar bilan bevosita ishlash. Operatsion tizim yadrosi kodining mashinali darajada yozilishi tizim ishining tezligi va xavfsizligini ta'minlashda hal qiluvchi ahamiyatga ega. Yuqori darajadagi tillar yordamida yozilgan kodlarda ba'zan kerakli darajada kechikishni kamaytirish yoki apparat xususiyatlarini to'liq nazorat qilish qiyin kechadi.

Drayverlar — qurilmalar va operatsion tizim o'rtasidagi interfeysdir. Har bir apparat qurilmaning o'ziga xos ish rejimi, boshqaruv registrari, va portlari bor. Drayver yozishda mashinali darajadagi dasturlar keng qo'llaniladi, chunki u apparat bilan aniq va tezkor muloqot qilish imkonini beradi. Masalan, disk drayverlari, grafik adapter drayverlari, tarmoq kartasi drayverlari kabi asosiy tizim qismlari mashinali kodlar yordamida qurilma registrariga yozuv va o'qish operatsiyalarini bajaradi. Drayverlarning mashinali kod bilan yozilishi: Resurslar va vaqtni tejaydi, kechikishlarni kamaytiradi. To'g'ridan-to'g'ri xotira va apparat portlariga kirishni ta'minlaydi. Yuqori darajadagi tillarda bajarilishi qiyin bo'lgan maxsus instruktsiyalarni ishlatadi. Bugungi kunda ko'pgina operatsion tizimlar (Windows, Linux, macOS) yadrosining pastki qatlamlarida mashinali kod ishlatiladi. Biroq, yuqori darajadagi tillarning rivojlanishi tufayli barcha kod mashinali darajada yozilmaydi - faqat maksimal samaradorlik talab qiladigan joylarda. Yangi texnologiyalar (masalan, virtualizatsiya, xavfsizlik protokollari) ham mashinali kodni o'rganish va

qo'llash zaruratini oshiradi, chunki apparatning yangi xususiyatlari to'g'ridan-to'g'ri boshqarilishi kerak bo'ladi.

Tahlil va natijalar

Mashinali darajadagi dasturlar tizimning pastki qatlamlarida ishlash samaradorligini sezilarli darajada oshiradi. Ushbu tahlilda bir nechta jihatlar ko'rib chiqildi. Operatsion tizim yadrosidagi mashinali kodlar protsessorning imkoniyatlaridan to'liq foydalanadi, shu bilan birga apparat va tizim o'rtasidagi kechikishlarni minimallashtiradi. Jarayonlar boshqaruvi va interruptlarni qayta ishlash kabi tezkor amallar mashinali kod yordamida maksimal darajada tez bajariladi. Misol uchun, interrupt handlerlar yuqori darajadagi tillarda yozilganda, kompilyator tomonidan qo'shimcha kod generatsiyasi kechikishni oshirishi mumkin, bu esa tizimning javob berish vaqtini uzaytiradi. Mashinali kodda esa bunday kechikishlar minimal bo'ladi. Yadro darajasidagi kodlarning mashinali darajada yozilishi tizimning xavfsizligini oshiradi. Chunki u apparat bilan bevosita muloqot qiladi va tashqi aralashuvlar va xatoliklar ehtimoli kamayadi. Shuningdek, noto'g'ri yozilgan yuqori darajadagi kod tizim ishini beqarorlashtirishi mumkin, bu esa katta muammolarga olib keladi. Mashinali kod yozish jarayoni murakkab va protsessor arxitekturasiga qattiq bog'langan. Bu esa dasturiy ta'minotni platformalararo ko'chirishni qiyinlashtiradi. Shu sababli, ko'pchilik tizim modullari yuqori darajadagi tillarda yoziladi va faqat eng muhim joylarda mashinali kod ishlatiladi. Tizim dasturchilari orasida olib borilgan tadqiqotlar shuni ko'rsatadiki, operatsion tizim yadrosida mashinali kod yordamida yozilgan interrupt handlerlar o'rtacha 15-20% tezroq ishlaydi, shu bilan birga tizimning umumiy javob berish vaqti sezilarli darajada qisqaradi. Drayverlar darajasida esa mashinali kod apparatning real vaqt rejimida ishlashini ta'minlab, qurilma ish faoliyatini barqarorlashtiradi.

Masalan, Intel x86 arxitekturasida uchun oddiy interrupt handler kodi quyidagicha yozilishi mumkin:

ISR:	
pusha	; barcha umumiy registrnlarni saqlash

mov al, 0x20	; End of Interrupt signal
out 0x20, al	; PIC ga signal yuborish
popa	; registrlarni tiklash
iretd	; interruptdan chiqish

Bu kod interrupt sodir bo'lganda protsessor holatini saqlaydi, signalni boshqaruv qurilmasiga yuboradi va dastur bajarilishini davom ettiradi. Bu jarayon yuqori darajadagi tillarda to'liq nazorat qilinishi qiyin va ko'p vaqt oladi.

Xulosa: Mashinali darajadagi dasturlar - bu kompyuterning apparat qismlari bilan bevosita ishlash imkonini beruvchi, maksimal samaradorlik va tezlikni ta'minlaydigan dasturlar hisoblanadi. Ular operatsion tizimlar va drayverlarning asosi bo'lib, tizim resurslarini to'liq nazorat qilish va boshqarish imkonini yaratadi. Shu sababli, mashinali kod tizimning eng pastki qatlamlarida, ya'ni yadro, apparat drayverlari va real vaqtli tizimlarda qo'llaniladi. Ularning asosiy afzalliklari -maksimal tezlik, samaradorlik va apparat bilan bevosita aloqada bo'lish qobiliyatidir. Bu esa tizimning ishonchliligini oshiradi va xavfsizlikni ta'minlashda muhim rol o'ynaydi. Masalan, operatsion tizim yadrosi mashinali kod yordamida tizimning barcha resurslarini samarali boshqaradi, foydalanuvchi va dasturlar o'rtasida xavfsiz interfeys hosil qiladi. Shu bilan birga, mashinali darajadagi dasturlarni yaratish juda murakkab jarayon bo'lib, yuqori darajadagi dasturlash tillariga qaraganda ko'p vaqt, bilim va tajriba talab qiladi. Bundan tashqari, ular protsessor arxitekturasini va platformaga qattiq bog'langanligi sababli, bitta platforma uchun yozilgan kodni boshqasida ishlatish qiyin. Bu esa ularning keng ko'lamda qo'llanilishini cheklaydi va dasturchilarning yuqori darajadagi tillarga o'tishiga sabab bo'ladi. Bugungi kunda mashinali kod ko'proq tizim dasturlari, apparat bilan muloqot qiluvchi drayverlar, real vaqtli tizimlar, shuningdek, maksimal tezlik talab qilinadigan ilmiy va sanoat ilovalarida qo'llaniladi. Yuqori darajadagi dasturlash tillari esa kundalik dasturlash jarayonlarini soddalashtirib, ko'p platformalarda ishlash imkonini beradi. Shunday qilib, mashinali darajadagi dasturlar o'zining murakkabligiga qaramay, kompyuter tizimlarining

barqarorligi va samaradorligini ta'minlashda, hamda zamonaviy texnologiyalarning asosini tashkil etishda o'chmas ahamiyatga ega. Ularning roli vaqt o'tishi bilan kamaymaydi, balki texnologiya rivojlanishi bilan yanada muhim va murakkab vazifalarni bajarishga moslashib boradi.

Foydalangan adabiyotlar

1. Tanenbaum, A. S. (2014). Modern Operating Systems (4th ed.). Pearson.
2. Stallings, W. (2018). Operating Systems: Internals and Design Principles (9th ed.). Pearson.
3. Patterson, D. A., & Hennessy, J. L. (2013). Computer Organization and Design: The Hardware/Software Interface (5th ed.). Morgan Kaufmann.
4. Bryant, R. E., & O'Hallaron, D. R. (2015). Computer Systems: A Programmer's Perspective (3rd ed.). Pearson.
5. Love, R. (2010). Linux Kernel Development (3rd ed.). Addison-Wesley.
6. Intel Corporation. (2019). Intel® 64 and IA-32 Architectures Software Developer's Manual.