

TIZIMLI DASTURLASH VA APPARAT VOSITALARI O‘RTASIDAGI INTEGRATSIYA

Umarov Bekzod Azizovich

Farg‘ona davlat universiteti

ubaumarov@gmail.com

Numonova Malika Abduxakim qizi

Farg‘ona davlat universiteti

numonovamalika38@gmail.com

Anotatsiya: Ushbu maqolada tizimli dasturlash va apparat vositalari o‘rtasidagi integratsiya jarayoni batafsil tahlil qilinadi. Maqolada apparat boshqaruvining asosiy usullari, qurilma drayverlari, tizim chaqiruvlari, interrupt mexanizmlari va past darajadagi dasturlash tillari orqali apparat va dasturiy ta‘minotni uyg‘unlashtirish yo‘llari ko‘rib chiqiladi. Shuningdek, real vaqtda tizimli dasturlashning apparat bilan integratsiyasidagi muhim jihatlar, uning sanoat, tibbiyot va transport sohalaridagi qo‘llanilishi haqida ma‘lumotlar beriladi.

Kalitso‘zlar: tizimli dasturlash, apparat boshqaruvi, qurilma drayverlari, tizim chaqiruvlari, interrupt, past darajadagi dasturlash, real vaqtda tizim, apparat integratsiyasi, operatsion tizim, RTOS, mikrokontroller, sanoat avtomatlashtirish, tibbiyot texnologiyalari, xavfsizlik, samaradorlik.

Abstract: This article provides a detailed analysis of the integration process between system programming and hardware devices. The article examines the main methods of hardware control, device drivers, system calls, interrupt mechanisms, and ways to synchronize hardware and software through low-level programming languages. Furthermore, important aspects of real-time system programming integration with hardware are discussed, along with its applications in industry, healthcare, and transportation sectors.

Keywords: system programming, hardware control, device drivers, system calls, interrupt, low-level programming, real-time system, hardware integration, operating system, RTOS, microcontroller, industrial automation, medical technologies, safety, efficiency.

Аннотация: В данной статье подробно рассматривается процесс интеграции системного программирования и аппаратных средств. В статье анализируются основные методы управления аппаратурой, драйверы устройств, системные вызовы, механизмы прерываний и способы синхронизации аппаратного и программного

обеспечения с использованием языков низкого уровня. Также обсуждаются важные аспекты интеграции системного программирования в реальном времени с аппаратурой, а также его применение в промышленности, медицине и транспортной сфере.

Ключевые слова: системное программирование, управление аппаратурой, драйверы устройств, системные вызовы, прерывание, программирование низкого уровня, система реального времени, интеграция аппаратуры, операционная система, RTOS, микроконтроллер, промышленная автоматизация, медицинские технологии, безопасность, эффективность.

Kirish

Tizimli dasturlash va apparat vositalari o‘rtasidagi integratsiya zamonaviy kompyuter tizimlarining samarali va barqaror ishlashi uchun muhim asoslardan biridir. Hozirgi kunda dasturiy ta’minot va apparatura bir-biridan ajralmas bo‘lib, ularning o‘zaro munosabati tizimning umumiy samaradorligi va ishlash tezligini belgilaydi. Tizimli dasturlash sohasida apparat bilan to‘g‘ridan-to‘g‘ri muloqot qilish imkonini beruvchi texnologiyalar va usullar doimo rivojlanib bormoqda, bu esa yangi imkoniyatlar bilan birga murakkabliklarni ham keltirib chiqarmoqda. Ushbu maqolada tizimli dasturlash va apparat vositalari o‘rtasidagi o‘zaro ta’sirlar, ularning integratsiya usullari hamda zamonaviy texnologiyalardagi roli tahlil qilinadi. Maqsadimiz – ushbu ikki sohaning uyg‘unlashuvi orqali kompyuter tizimlarining samaradorligini oshirish yo‘llarini o‘rganish va tizimli dasturlash asosidagi apparat boshqaruvi mexanizmlarini yanada chuqurroq tushunishdir. Shu bilan birga, maqola davomida real vaqtda tizimli dasturlash hamda qurilma drayverlari kabi muhim yo‘nalishlarga ham e’tibor qaratiladi. Tizimli dasturlash va apparat integratsiyasi sohasidagi zamonaviy tendensiyalarni ko‘rib chiqish, shuningdek, kelajakdagi istiqbollarni aniqlash orqali ushbu yo‘nalishda ilmiy va amaliy yutuqlarga erishish imkoniyatlari muhokama qilinadi.

Tizimli dasturlash – bu kompyuterning apparat qismlari bilan yaqin va bevosita aloqada bo‘lish uchun mo‘ljallangan dasturlash yo‘nalishidir. U operatsion tizimlar, qurilma drayverlari, kompilyatorlar va boshqa past darajadagi dasturlarni yaratishda qo‘llaniladi. Tizimli dasturlashda dasturchi tizim resurslarini boshqarish, apparat vositalari bilan o‘zaro ta’sirda bo‘lish va samarali ishlash uchun kod yozadi. Bu jarayon yuqori darajadagi dasturlashdan farq qiluvchi o‘ziga xosliklarga ega bo‘lib, unda dasturchi protsessorning registrarlari, xotira manzillari va tizim chaqiruvlari kabi texnik jihatlar bilan ishlaydi. Shunday qilib, tizimli dasturlash kompyuterning asosiy resurslarini boshqarish imkonini beruvchi, juda aniqlik va puxta bilim talab qiluvchi jarayondir.

Apparat vositalari yoki hardware – bu kompyuter tizimining fizik qismlari bo‘lib, ular dasturiy ta‘minot tomonidan boshqariladi. Asosiy apparat komponentlari protsessor (CPU), operativ xotira (RAM), kirish-chiqish qurilmalari (klaviatura, sichqoncha, printer kabi), saqlash qurilmalari (qattiq disklar, SSD), va boshqalar hisoblanadi. Protsessor kompyuterning markaziy hisoblash qismidir, u dasturlarni bajarish uchun zarur bo‘lgan barcha arifmetik va mantiqiy operatsiyalarni amalga oshiradi. Operativ xotira esa vaqtinchalik ma‘lumotlar va kodlarni saqlaydi, protsessor esa ularga tezkor kirishni ta‘minlaydi. Qurilma drayverlari esa apparat vositalari va operatsion tizim o‘rtasidagi vositachilar bo‘lib, ular apparatni dastur tomonidan boshqarilishini ta‘minlaydi. Tizimli dasturlash va apparat vositalari o‘rtasidagi o‘zaro ta‘siralar dasturiy ta‘minotning samarali ishlashi uchun juda muhimdir. Dasturchi apparat bilan to‘g‘ridan-to‘g‘ri muloqot qilish uchun tizim chaqiruvlari, interrupt (to‘xtatish) mexanizmlari, hamda qurilma drayverlarini yaratadi. Masalan, qurilma drayveri kompyuterga ulangan printer yoki disk drayveri kabi apparatni boshqarish uchun yozilgan maxsus dasturiy ta‘minotdir. Ushbu drayverlar apparatni tizim resurslariga bog‘lab, operatsion tizimga ushbu qurilmalardan foydalanish imkoniyatini beradi. Shunday qilib, tizimli dasturlash yordamida apparatning imkoniyatlari to‘liq ochiladi va undan samarali foydalanish ta‘minlanadi.

Mikrokontrollerlar – tizimli dasturlash va apparat vositalari integratsiyasining yana bir muhim jihati hisoblanadi. Mikrokontrollerlar kichik o‘lchamdagi, o‘z ichida protsessor, xotira va kirish-chiqish portlarini jamlagan maxsus apparat qismlaridir. Ular ko‘pincha sanoat, avtomatlashtirish, robototexnika va boshqa ko‘plab sohalarda qo‘llaniladi. Mikrokontrollerlarni dasturlashda tizimli dasturlash tamoyillari asosiy rol o‘ynaydi, chunki dasturchi bevosita apparatning turli qismlarini boshqarishi, real vaqtda ishni ta‘minlashi va samaradorlikni oshirishi zarur. Shu bois mikrokontroller dasturlashida apparat va dasturiy ta‘minot o‘rtasidagi integratsiya eng yuqori darajada amalga oshiriladi. Umuman olganda, tizimli dasturlash va apparat vositalari tushunchalari bir-birini to‘ldiruvchi elementlar hisoblanadi. Tizimli dasturlash apparat imkoniyatlaridan samarali foydalanishni ta‘minlab, kompyuter tizimlarining barqaror va tezkor ishlashini kafolatlaydi. Ushbu ikki sohaning uyg‘unlashuvi hozirgi zamon texnologiyalarining asosini tashkil etib, yangi imkoniyatlar va yechimlarni yaratishda muhim ahamiyatga ega. Shu sababli, tizimli dasturlash va apparat vositalari o‘rtasidagi integratsiyani chuqur o‘rganish va rivojlantirish ilmiy va amaliy jihatdan katta ahamiyat kasb etadi.

Tizimli dasturlashning asosiy vazifalaridan biri — apparat vositalarini samarali va ishonchli boshqarishdir. Kompyuterning apparat qismlari o‘zaro aloqada bo‘lib, ularni nazorat qilish uchun maxsus dasturiy ta‘minot zarur. Tizimli dasturlash orqali apparat boshqaruvi asosan qurilma drayverlari, tizim chaqiruvlari va past darajadagi dasturlash

metodlari orqali amalga oshiriladi. Ushbu usullar bir-birini to‘ldiradi va zamonaviy kompyuter tizimlarining uzluksiz ishlashini ta‘minlaydi. Tizimli dasturlash darajasida qurilma bilan ishlash jarayonida dasturchi apparatning o‘ziga xos xususiyatlarini hisobga olgan holda, kod yozishi lozim. Bu kod ko‘pincha apparatga bevosita kirish imkonini beruvchi portlar, registrlar va xotira manzillari bilan ishlaydi. Qurilma drayverlari - apparatni boshqarishdagi eng muhim vositalardan biridir. Drayver - bu operatsion tizim va apparat qurilmasi o‘rtasida vositachi vazifasini bajaruvchi dasturiy ta‘minotdir. U apparatning xususiyatlarini operatsion tizimga tushunarli qilib tarjima qiladi va qurilmaning samarali ishlashini ta‘minlaydi. Masalan, printer, grafik karta yoki tarmoq adapteri kabi apparatlar uchun alohida drayverlar mavjud bo‘lib, ular qurilmani to‘liq nazorat qiladilar. Drayverlar tizim chaqiruvlari orqali ishlaydi, ular yordamida yuqori darajadagi dasturlar apparat bilan bog‘lanadi. Tizimli dasturchi drayverni yaratishda apparatning texnik tavsiflarini, uning protokollarini va boshqaruv registrlarini mukammal tushunishi kerak. Drayverlar apparatga buyruqlar yuboradi, ma‘lumotlarni qabul qiladi va xatoliklarni aniqlaydi, bu esa tizimning barqaror va samarali ishlashiga yordam beradi.

Tizim chaqiruvlari (system calls) — bu operatsion tizim yadrosi va foydalanuvchi dasturi o‘rtasidagi interfeysdir. Tizimli dasturlashda tizim chaqiruvlari orqali dasturlar apparat resurslarini boshqarishga ruxsat oladi. Masalan, fayl tizimi bilan ishlash, tarmoq ulanishlari, qurilmalarni boshqarish tizim chaqiruvlari orqali amalga oshiriladi. Tizim chaqiruvlari yordamida dasturchi qurilmalarni bevosita boshqarish imkoniyatiga ega bo‘ladi, ammo bu jarayon operatsion tizim tomonidan qat‘iy nazorat ostida bo‘ladi, chunki to‘g‘ridan-to‘g‘ri apparatga kira olish tizimning xavfsizligi va barqarorligini ta‘minlaydi. Tizim chaqiruvlari orqali apparatga buyruq berish va ma‘lumotlarni uzatish juda muhim bo‘lib, ular apparat va dasturiy ta‘minot o‘rtasidagi muvozanatni saqlaydi. Past darajadagi dasturlash tillari, xususan assembler, tizimli dasturlashda apparatni boshqarish uchun keng qo‘llaniladi. Assembler tilida yozilgan dasturlar bevosita protsessor registrari va apparat manzillariga murojaat qiladi. Bu dasturlash usuli tizimli darajada juda tezkor va samarali hisoblanadi, chunki u yuqori darajadagi tillardan farqli o‘laroq, bevosita apparatga yaqin bo‘ladi. Misol uchun, qurilmaning kontrol registrlarini o‘zgartirish, signal uzatish, interrupt (to‘xtatish) mexanizmlarini boshqarish assembler orqali ancha oson amalga oshiriladi. Shu bilan birga, assemblerda dasturlash murakkabroq va ko‘proq xatoliklar yuzaga kelishi mumkin, shuning uchun u odatda maxsus tizimli funksiyalar va drayverlar yaratishda ishlatiladi.

Interruptlar (to‘xtatishlar) apparat boshqaruvida muhim rol o‘ynaydi. Bu — apparat qurilmalari tizimga signal yuborib, o‘zlariga e‘tibor qaratish uchun ishlatiladigan mexanizm. Masalan, klaviatura tugmachasini bosganingizda, u protsessorga signal yuboradi

va shu orqali protsessor ushbu voqea haqida xabardor bo‘ladi. Tizimli dasturlashda interrupt handlerlar (to‘xtatishlarni qayta ishlovchi dasturlar) yoziladi, ular signalni qabul qilib, kerakli jarayonlarni bajaradi. Bu usul apparat bilan samarali va kechikishsiz muloqot qilish imkonini beradi. Interruptlar yordamida tizim resurslari samarali boshqariladi va apparat qurilmalaridan to‘liq foydalanish ta‘minlanadi. Mikrokontrollerlar va ularning dasturlanishi tizimli dasturlash orqali apparat boshqarishning yana bir muhim jihatidir. Mikrokontrollerlar kichik, o‘zida protsessor, xotira va periferik qurilmalarni jamlagan maxsus apparat bloklaridir. Ularni dasturlashda bevosita apparat registrarlari va qurilmalarni boshqarish imkonini beruvchi tillar va vositalar qo‘llaniladi. Mikrokontrollerlarda ko‘pincha real vaqtda tizimli dasturlash tamoyillari qo‘llaniladi, chunki ko‘plab dasturlar o‘z vaqtida javob qaytarishi lozim. Mikrokontroller dasturlashda tizimli dasturchi apparatning har bir funksiyasini aniq bilishi va uni samarali boshqarishni ta‘minlashi kerak. Ushbu yo‘nalish sanoat avtomatlashtirish, robototexnika, kiberjismaniy tizimlar kabi sohalarda keng qo‘llaniladi. Shuningdek, zamonaviy tizimli dasturlashda apparat boshqaruvining yana bir yo‘li - virtualizatsiya texnologiyalari hisoblanadi. Virtualizatsiya yordamida bitta apparat platformasida bir nechta mustaqil tizimlar ishlashi mumkin. Bu jarayonda tizimli dasturlash apparat resurslarini qat‘iy taqsimlash, ularni himoya qilish va samarali boshqarishni ta‘minlaydi. Virtual mashinalar va konteyner texnologiyalari apparat va dasturiy ta‘minotni yanada qattiqroq integratsiyalashga imkon beradi, bu esa ko‘p foydalanuvchi va ko‘p dasturli muhitlarda samaradorlikni oshiradi. Umuman olganda, tizimli dasturlash orqali apparat vositalarini boshqarish ko‘plab usullar va texnikalarni o‘z ichiga oladi. Har bir usul o‘ziga xos vazifa va imkoniyatlarga ega bo‘lib, ular birgalikda zamonaviy kompyuter tizimlarining barqaror, tezkor va xavfsiz ishlashini ta‘minlaydi. Qurilma drayverlari, tizim chaqiruvlari, interruptlar, past darajadagi dasturlash va mikrokontroller dasturlash - bularning barchasi apparat bilan muloqotni ta‘minlaydigan asosiy mexanizmlar hisoblanadi. Ushbu mexanizmlarni puxta o‘rganish va qo‘llash tizimli dasturchilardan yuqori darajada bilim va malaka talab etadi, chunki apparat boshqaruvi tizimning butun ishlash samaradorligini belgilovchi omillardan biridir. Shu sababli, tizimli dasturlashda apparat vositalarini boshqarish usullarini chuqur tushunish zamonaviy texnologiyalar sohasida muvaffaqiyatli ishlash uchun muhim shartdir.

Real vaqtda tizimli dasturlash va apparat integratsiyasi zamonaviy kompyuter tizimlari va sanoat avtomatik boshqaruv sohasida katta ahamiyat kasb etadi. Real vaqtda tizimlar o‘z ish faoliyatida belgilangan vaqt chegaralari ichida aniq va ishonchli natijalarni taqdim etishi shart bo‘lgan dasturiy va apparat komplekslaridir. Bunday tizimlarda kechikishlar va noaniqliklar qabul qilinmaydi, chunki ular xavfsizlik, ishlab chiqarish jarayonlari, tibbiyot, aviatsiya va transport kabi muhim sohalarda qo‘llaniladi. Real vaqtda tizimli dasturlash esa,

aynan shu talablarni qondirish uchun mo‘ljallangan dasturlash yondoshuvlar majmuasidir. U apparat vositalari bilan aniq sinxronlashgan holda ishlaydi va tizim resurslarini o‘z vaqtida boshqarishni ta‘minlaydi.

Real vaqtda operatsion tizimlar (RTOS) — real vaqtda tizimli dasturlashning muhim qismi bo‘lib, ular apparat platformasida samarali ishlash uchun zarur vositalarni taqdim etadi. RTOSlarning asosiy vazifasi — vaqtni qat’iy nazorat qilish, vazifalarni prioritetga ko‘ra taqsimlash, to‘xtatishlarni (interrupt) boshqarish va resurslarga kirishni muvofiqlashtirishdir. Ushbu tizimlar yordamida dasturchi dasturlarini apparat bilan yaqin integratsiyada ishlab chiqadi, bunda har bir vazifa o‘z vaqtida bajarilishi kafolatlanadi. RTOS apparatning real signal va voqealariga tezkor javob berish imkoniyatiga ega bo‘lib, bu sanoat va harbiy sohalarda juda katta afzallik beradi. Real vaqtda tizimli dasturlashda apparat integratsiyasi muhim texnik jihatlar bilan belgilanadi. Masalan, interrupt mexanizmlari yordamida apparat qurilmalari tizimga voqealar haqida signal yuboradi va RTOS bu signalga tezda javob qaytaradi. Bu jarayon apparat va dasturiy ta‘minot o‘rtasidagi o‘zaro bog‘lanishni kuchaytiradi va tizimning samaradorligini oshiradi. Shuningdek, real vaqtda tizimlarda qurilma drayverlari va apparat registrlariga bevosita kirish o‘ta muhimdir, chunki bu kechikishlarni kamaytiradi va apparat imkoniyatlaridan maksimal darajada foydalanishga imkon beradi. Tizimli dasturlash orqali yozilgan kodlar apparatning real holatini doimiy nazorat qilib boradi, shuningdek, zarur bo‘lganda vaqtinchalik o‘zgartirishlar kiritadi. Real vaqtda tizimli dasturlash va apparat integratsiyasining qo‘llanilish sohalari juda keng. Masalan, avtomobil sanoatida haydovchisiz avtomobillar, avtonom boshqaruv tizimlari real vaqtda apparat va dasturiy ta‘minot integratsiyasiga tayanadi. Tibbiyotda esa yurakni kuzatish qurilmalari, sun‘iy nafas olish apparatlari kabi murakkab tizimlar real vaqtda ishlaydi va ularning ishonchligi bemorning hayoti uchun juda muhimdir. Shu bilan birga, sanoat avtomatlashtirishda ishlab chiqarish liniyalarining monitoringi va boshqaruvi real vaqtda tizimli dasturlash yordamida amalga oshiriladi. Ushbu sohalarda tizimlarning real vaqt ichida to‘g‘ri ishlashi apparat va dasturiy ta‘minotning mukammal integratsiyasiga bog‘liqdir.

Real vaqtda tizimli dasturlash va apparat integratsiyasi zamonaviy texnologiyalarda barqaror va samarali ishlashni ta‘minlovchi muhim omil hisoblanadi. Bu integratsiya yordamida murakkab tizimlar o‘z vazifalarini aniq vaqt doirasida bajaradi, bu esa xavfsizlik va unumdorlikni oshiradi. Kelajakda real vaqtda tizimli dasturlash sohasida yangi texnologiyalar, masalan, sun‘iy intellekt va kiber-fizik tizimlar bilan integratsiya yanada rivojlanishi kutilmoqda. Shu bois, ushbu yo‘nalishdagi ilmiy tadqiqotlar va amaliy ishlanmalar soha taraqqiyotining poydevorini tashkil etadi.

Xulosa

Tizimli dasturlash va apparat vositalari o‘rtasidagi integratsiya zamonaviy kompyuter tizimlari va texnologiyalarining samarali ishlashining asosiy tamoyillaridan biridir. Ushbu integratsiya orqali dasturiy ta’minot apparat resurslarini to‘liq boshqarish imkoniga ega bo‘lib, tizimlarning tezkorligi, barqarorligi va xavfsizligini ta’minlaydi. Qurilma drayverlari, tizim chaqiruvlari, interruptlar va past darajadagi dasturlash tillari apparat va dasturiy ta’minotni mukammal uyg‘unlashtirishda muhim vositalar hisoblanadi. Ayniqsa, real vaqtda tizimli dasturlash sohasida bu integratsiya o‘zining o‘ziga xos ahamiyatini ko‘rsatadi, chunki u vaqt chegaralariga qat’iy rioya qilgan holda, tizim resurslarini samarali boshqarishga xizmat qiladi. Bugungi kunda sanoat avtomatlashtirish, tibbiyot, transport va boshqa ko‘plab sohalarda tizimli dasturlash va apparat integratsiyasi yangi texnologiyalarni rivojlantirish hamda murakkab tizimlarni barqaror ishlashini ta’minlashda muhim omil hisoblanadi. Shuning uchun, ushbu sohani chuqur o‘rganish va rivojlantirish zamonaviy texnologiyalarning kelajagi uchun katta ahamiyatga ega.

Foydalanilgan adabiyotlar:

1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts*. Wiley.
2. Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems*. Pearson.
3. Stallings, W. (2014). *Operating Systems: Internals and Design Principles*. Pearson.
4. J. L. Hennessy & D. A. Patterson (2017). *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann.
5. Bovet, D. P., & Cesati, M. (2005). *Understanding the Linux Kernel*. O'Reilly Media.
6. Labrosse, J. J. (2002). *MicroC/OS-II: The Real-Time Kernel*. CMP Books.
7. Liu, J. W. S. (2000). *Real-Time Systems*. Prentice Hall.
8. Michael J. Pont (2011). *Real-Time Embedded Systems*. CRC Press.
9. Valvano, J. W. (2016). *Embedded Microcomputer Systems: Real Time Interfacing*. Cengage Learning.