

XOTBRANING SEGMENT-SAHIFALI TAQSIMLASH.

*Tojimamatov Israiljon Nurmamatovich**Farg'ona Davlat Universiteti*[*israiltojimatov@gmail.com*](mailto:israiltojimatov@gmail.com)*Qurbonova Mohidil Mirqo'zi qizi**Farg'ona Davlat Universiteti 2-kurs talabasi*[*mohidil996@gmail.com*](mailto:mohidil996@gmail.com)

Annotatsiya: Ushbu maqolada operatsion tizimlarda xotirani boshqarishning gibridd usuli bo'lgan segment-sahifali taqsimlash mexanizmi batafsil ko'rib chiqiladi. Maqolada segmentatsiya va sahifalash afzalliklarini birlashtirgan ushbu yondashuvning asosiy tamoyillari, ishlash tartibi, afzalliklari va kamchiliklari tahlil qilinadi. Xususan, mantiqiy manzillarni fizik manzillarga o'tkazish jarayoni, segment jadvallari va sahifa jadvallarining roli, ichki va tashqi fragmentatsiya muammolariga ta'siri yoritiladi. Shuningdek, segment-sahifali taqsimlashning sof segmentatsiya va sof sahifalash usullaridan farqlari va ustunliklari muhokama qilinadi.

Kalit so'zlar: xotirani boshqarish, segment-sahifali taqsimlash, segmentatsiya, sahifalash, operatsion tizimlar, virtual xotira, mantiqiy manzil, fizik manzil, segment jadvali, sahifa jadvali, fragmentatsiya.

Аннотация: В данной статье подробно рассматривается механизм сегментно-страничного распределения памяти – гибридный подход к управлению памятью в операционных системах. В статье анализируются основные принципы, порядок работы, преимущества и недостатки данного подхода, сочетающего достоинства сегментации и страничной организации. В частности, освещаются процесс преобразования логических адресов в

физические, роль таблиц сегментов и таблиц страниц, а также влияние на проблемы внутренней и внешней фрагментации. Кроме того, обсуждаются отличия и преимущества сегментно-страничного распределения по сравнению с методами чистой сегментации и чистой страничной организации.

Ключевые слова: управление памятью, сегментно-страничное распределение, сегментация, страничная организация, операционные системы, виртуальная память, логический адрес, физический адрес, таблица сегментов, таблица страниц, фрагментация.

Annotation: This article provides a detailed examination of the segmented-paged memory allocation mechanism, a hybrid approach to memory management in operating systems. The article analyzes the fundamental principles, operational procedures, advantages, and disadvantages of this approach, which combines the benefits of segmentation and paging. Specifically, it elucidates the process of translating logical addresses to physical addresses, the roles of segment tables and page tables, and its impact on internal and external fragmentation issues. Furthermore, the differences and advantages of segmented-paged allocation compared to pure segmentation and pure paging methods are discussed.

Keywords: memory management, segmented-paged allocation, segmentation, paging, operating systems, virtual memory, logical address, physical address, segment table, page table, fragmentation.

Kirish

Xotirani boshqarish zamonaviy operatsion tizimlarning eng muhim va murakkab vazifalaridan biridir. Samarali xotira boshqaruvi tizimning umumiy unumdorligi, barqarorligi va bir vaqtning o'zida bir nechta dasturlarning ishlashini ta'minlash uchun hal qiluvchi ahamiyatga ega. Tarixan xotirani boshqarishning turli usullari, jumladan, oddiy bo'laklash (fixed partitioning), dinamik bo'laklash (dynamic partitioning), sahifalash (paging) va segmentatsiya (segmentation) kabi yondashuvlar ishlab

chiqilgan va qo'llanilgan. Ushbu usullarning har biri o'zining afzalliklari va kamchiliklariga ega.

Sof sahifalash xotirani bir xil o'lchamdagi kichik bloklarga – sahifalarga bo'lish orqali tashqi fragmentatsiyani bartaraf etadi, ammo dasturning mantiqiy tuzilishini hisobga olmaydi va ichki fragmentatsiyaga olib kelishi mumkin. Boshqa tomondan, sof segmentatsiya xotirani dasturning mantiqiy birliklariga (masalan, kod segmenti, ma'lumotlar segmenti, stek segmenti) mos keladigan o'zgaruvchan o'lchamdagi segmentlarga bo'ladi. Bu dasturchi uchun qulaylik yaratadi va himoya mexanizmlarini amalga oshirishni osonlashtiradi, ammo tashqi fragmentatsiyaga moyil bo'ladi.

Ushbu ikki asosiy yondashuvning kamchiliklarini bartaraf etish va ularning afzalliklarini birlashtirish maqsadida gibrid usullar ishlab chiqilgan. Shunday usullardan biri – xotiraning segment-sahifali taqsimlanishidir. Ushbu yondashuvda xotira avval segmentlarga bo'linadi, so'ngra har bir segment o'z navbatida sahifalarga ajratiladi. Bu esa ham dasturning mantiqiy tuzilishini saqlab qolishga, ham xotiradan samarali foydalanishga imkon beradi.

Ushbu maqolaning maqsadi xotiraning segment-sahifali taqsimlanish mexanizmini chuqur tahlil qilish, uning ishlash prinsiplarini, afzalliklari va kamchiliklarini yoritib berishdan iborat. Maqolada mantiqiy manzillarni fizik manzillarga o'tkazish jarayoni, segment va sahifa jadvallarining roli, fragmentatsiya muammolariga ta'siri hamda ushbu usulning boshqa xotira boshqaruv usullari bilan taqqoslanishi ko'rib chiqiladi. Ushbu bilimlar operatsion tizimlar arxitekturasini va xotira boshqaruvi bilan shug'ullanuvchi mutaxassislar uchun muhim ahamiyat kasb etadi.

Asosiy qism

Xotiraning segment-sahifali taqsimlanishi operatsion tizimlarda xotirani boshqarishning murakkab, ammo samarali usullaridan biri bo'lib, u ham segmentatsiyaning, ham sahifalashning afzalliklarini o'zida mujassamlashtirishga

harakat qiladi. Ushbu yondashuvda dasturning mantiqiy manzillar fazosi avval segmentlarga, so'ngra har bir segment o'z navbatida bir xil o'lchamdagi sahifalarga bo'linadi. Bu gibrid sxema xotiradan foydalanish samaradorligini oshirish bilan birga dasturning mantiqiy tuzilishini ham saqlab qolishga imkon beradi.

Segment-sahifali taqsimlashning ishlash mexanizmi

Segment-sahifali xotira boshqaruvida har bir dastur bir yoki bir nechta segmentlardan tashkil topgan deb qaraladi. Har bir segment dasturning mantiqiy birligini ifodalaydi, masalan, kod segmenti, ma'lumotlar segmenti, stek segmenti va hokazo. Sof segmentatsiyadan farqli o'laroq, bu segmentlar o'zgaruvchan o'lchamda bo'lishi mumkin, ammo ular fizik xotiraga joylashtirilishidan oldin sahifalarga bo'linadi. Barcha sahifalar bir xil, qat'iy belgilangan o'lchamga ega bo'ladi, bu esa fizik xotirani boshqarishni osonlashtiradi.

Tutorialspoint (n.d.) ta'kidlaganidek, ikkita asosiy yondashuv mavjud: **sahifalangan segmentatsiya (paged segmentation)** va **segmentlangan sahifalash (segmented paging)**. Garchi nomlari o'xshash bo'lsa-da, ular bir oz farq qiladi:

- **Sahifalangan segmentatsiya (Paged Segmentation):** Bunda xotira avval o'zgaruvchan o'lchamdagi segmentlarga bo'linadi. Keyin har bir segment qat'iy belgilangan o'lchamdagi sahifalarga ajratiladi. Mantiqiy manzil segment raqami va segment ichidagi siljishdan iborat bo'ladi. Bu siljish keyinchalik sahifa raqami va sahifa ichidagi siljishga aylantiriladi.

- **Segmentlangan sahifalash (Segmented Paging):** Bunda xotira avval qat'iy belgilangan o'lchamdagi sahifalarga bo'linadi. Keyin bu sahifalar mantiqiy segmentlarga guruhlanishi mumkin. Bu yondashuv kamroq tarqalgan va ko'pincha "sahifalangan segmentatsiya" atamasi kengroq ma'noda ikkala holatni ham qamrab olishi mumkin deb tushuniladi. Asosiy e'tibor odatda birinchi yondashuvga qaratiladi, ya'ni segmentlar sahifalarga bo'linadi.

Manzillarni o'tkazish (Address Translation)

Segment-sahifali taqsimlashda mantiqiy manzilni fizik manzilga o'tkazish ikki bosqichli jarayonni o'z ichiga oladi:

1. **Segment jadvali (Segment Table):** Mantiqiy manzil odatda uch qismdan iborat bo'ladi: segment raqami (s), sahifa raqami (p) va sahifa ichidagi siljish (d). Dastlab, segment raqami (s) segment jadvalini indekslash uchun ishlatiladi. Segment jadvalidagi har bir yozuv (segment deskriptori) mos keladigan segment haqida ma'lumotlarni, jumladan, segmentning sahifa jadvalining bazaviy manzilini va segmentning chegarasini (o'lchamini) o'z ichiga oladi. Siljish segment chegarasidan oshib ketmaganligi tekshiriladi.

2. **Sahifa jadvali (Page Table):** Segment jadvalidan olingan sahifa jadvalining bazaviy manziliga sahifa raqami (p) qo'shilib, kerakli sahifa yozuviga murojaat qilinadi. Sahifa jadvalidagi har bir yozuv mos keladigan sahifaning fizik xotiradagi ramka (frame) raqamini o'z ichiga oladi. Ushbu ramka raqami sahifa ichidagi siljish (d) bilan birlashtirilib, yakuniy fizik manzil hosil qilinadi.

Bu jarayon GeeksforGeeks (2024) maqolasida keltirilgan sof sahifalash va sof segmentatsiyadagi manzillarni o'tkazish jarayonlarining kombinatsiyasi sifatida qaralishi mumkin. Segmentatsiya dasturning mantiqiy tuzilishini ta'minlasa, sahifalash esa xotiradan samarali foydalanish va fragmentatsiyani kamaytirishga yordam beradi.

Segment-sahifali taqsimlashning afzalliklari

Segment-sahifali xotira boshqaruvi bir qator muhim afzalliklarga ega:

- **Mantiqiy tuzilmani saqlash:** Segmentatsiya elementini saqlab qolish orqali dasturning mantiqiy birliklari (kod, ma'lumotlar, stek) alohida himoyalaniishi va ulashilishi (sharing) mumkin. Bu dasturchilar uchun qulaylik yaratadi.

• **Fragmentatsiyani kamaytirish:** Har bir segment sahifalarga bo'linganligi sababli, tashqi fragmentatsiya muammosi sezilarli darajada kamayadi. Sahifalar kichik va bir xil o'lchamda bo'lgani uchun xotirada bo'sh joylarni topish va ulardan foydalanish osonlashadi. Ichki fragmentatsiya esa faqat har bir segmentning oxirgi sahifasida yuzaga kelishi mumkin.

• **Moslashuvchanlik:** Segmentlar o'zgaruvchan o'lchamda bo'lishi mumkin, bu esa dasturning turli ehtiyojlariga moslashish imkonini beradi.

• **Himoya va ulashish imkoniyatlari:** Segment darajasida himoya bitlari (o'qish, yozish, bajarish huquqlari) va ulashish mexanizmlarini qo'llash osonlashadi. Turli jarayonlar bir xil kod segmentini ulashishi mumkin, bu esa xotirani tejashga yordam beradi.

• **Dinamik o'sish:** Segmentlar dinamik ravishda o'sishi yoki qisqarishi mumkin (ma'lum darajada), bu esa dasturning xotira ehtiyojlarining o'zgarishiga moslashishga yordam beradi.

Segment-sahifali taqsimlashning kamchiliklari

Afzalliklariga qaramay, segment-sahifali taqsimlashning ba'zi kamchiliklari ham mavjud:

• **Murakkablik:** Bu yondashuv sof sahifalash yoki sof segmentatsiyaga qaraganda ancha murakkab. Manzillarni o'tkazish jarayoni ikki bosqichli bo'lib, ikkita jadvalga (segment jadvali va sahifa jadvali) murojaat qilishni talab etadi. Bu esa manzillarni o'tkazish vaqtini oshirishi mumkin.

• **Qo'shimcha xarajatlar (Overhead):** Ikkita jadvalni saqlash va ularga murojaat qilish uchun qo'shimcha xotira va apparat ta'minoti (masalan, TLB – Translation Lookaside Buffer) talab etiladi. Bu tizimning umumiy xarajatlarini oshirishi mumkin.

• **Ichki fragmentatsiya:** Garchi tashqi fragmentatsiya kamaytirilgan bo'lsa-da, har bir segmentning oxirgi sahifasida ichki fragmentatsiya yuzaga kelishi mumkin. Agar segmentlar soni ko'p bo'lsa, bu umumiy ichki fragmentatsiyani sezilarli darajada oshirishi mumkin.

• **Manzillarni o'tkazish vaqti:** Ikkita jadvalga murojaat qilish zarurati tufayli manzillarni o'tkazish vaqti sof sahifalashga qaraganda uzoqroq bo'lishi mumkin. Biroq, TLB kabi kesh mexanizmlari bu muammoni yumshatishga yordam beradi.

Sof segmentatsiya va sof sahifalash bilan taqqoslash

Sof sahifalash bilan taqqoslaganda:

• **Afzalliklari:** Segment-sahifali taqsimlash dasturning mantiqiy tuzilishini yaxshiroq aks ettiradi va segment darajasida himoya hamda ulashishni osonlashtiradi.

• **Kamchiliklari:** Manzillarni o'tkazish jarayoni murakkabroq va sekinroq bo'lishi mumkin (TLB hisobga olinmaganda).

Sof segmentatsiya bilan taqqoslaganda:

• **Afzalliklari:** Segment-sahifali taqsimlash tashqi fragmentatsiyani deyarli yo'q qiladi, chunki segmentlar sahifalarga bo'linadi va bu sahifalar xotiraning istalgan bo'sh ramkalariga joylashtirilishi mumkin.

• **Kamchiliklari:** Ichki fragmentatsiya muammosi (segmentning oxirgi sahifasida) paydo bo'ladi va manzillarni o'tkazish mexanizmi murakkablashadi.

GeeksforGeeks (2024) maqolasida keltirilganidek, sahifalashda dastur bir xil o'lchamdagi sahifalarga bo'linadi va operatsion tizim bu jarayon uchun mas'uldir, segmentatsiyada esa dastur o'zgaruvchan o'lchamdagi segmentlarga bo'linadi va bu ko'pincha kompilyator tomonidan amalga oshiriladi. Segment-sahifali taqsimlash ushbu ikki yondashuvni birlashtirib, ularning ijobiy tomonlarini olishga intiladi.

Amaliy qo'llanilishi

Segment-sahifali xotira boshqaruvi ba'zi tarixiy va zamonaviy operatsion tizimlarda qo'llanilgan yoki qo'llanilmoqda. Masalan, Intel x86 arxitekturasida (ayniqsa, 32-bitli rejimda) segmentatsiya va sahifalashni birgalikda qo'llab-quvvatlaydi. Bu arxitekturalarda segmentlar himoya va manzillar fazosini ajratish uchun ishlatilsa, sahifalash esa virtual xotirani amalga oshirish va xotiradan samarali foydalanish uchun ishlatiladi.

Biroq, ko'plab zamonaviy 64-bitli operatsion tizimlar (masalan, Windows, Linux, macOS ning yangi versiyalari) asosan

Xulosa

Xotiraning segment-sahifali taqsimlanishi operatsion tizimlarda xotirani boshqarishning muhim gibridd yondashuvlaridan biri sifatida namoyon bo'ladi. Ushbu usul segmentatsiyaning mantiqiy afzalliklarini (dastur tuzilishini aks ettirish, himoya va ulashish qulayligi) sahifalashning fizik afzalliklari (tashqi fragmentatsiyani bartaraf etish, xotiradan samarali foydalanish) bilan birlashtirishga intiladi. Maqolada ko'rib chiqilganidek, bu yondashuvda dasturning mantiqiy manzillar fazosi avval segmentlarga, so'ngra har bir segment sahifalarga bo'linadi, bu esa manzillarni o'tkazishda ikki bosqichli jarayonni (segment jadvali va sahifa jadvali orqali) talab etadi.

Segment-sahifali taqsimlashning asosiy yutuqlari qatoriga mantiqiy tuzilmani saqlagan holda fragmentatsiyani kamaytirish, moslashuvchanlikni ta'minlash hamda himoya va ulashish imkoniyatlarini kengaytirish kiradi. Biroq, bu usul o'zining murakkabligi, manzillarni o'tkazishdagi qo'shimcha xarajatlar va vaqt sarfi hamda segmentlarning oxirgi sahifalarida yuzaga kelishi mumkin bo'lgan ichki fragmentatsiya kabi kamchiliklardan xoli emas. TLB kabi apparat yordami bu xarajatlarni kamaytirishda muhim rol o'ynaydi.

Sof segmentatsiya va sof sahifalash bilan taqqoslaganda, segment-sahifali yondashuv muayyan muvozanatni taklif etadi. U sof segmentatsiyaning tashqi fragmentatsiya muammosini hal qilsa, sof sahifalashning dastur mantiqg'iga befarqligini bartaraf etadi. Intel x86 kabi ba'zi arxitekturalarda tarixan qo'llanilgan bo'lsa-da, zamonaviy 64-bitli operatsion tizimlar ko'pincha soddalashtirilgan yondashuv sifatida asosan sahifalashga tayanadi va segmentatsiyadan cheklangan darajada (masalan, himoya uchun) foydalanadi yoki umuman foydalanmaydi.

Xulosa qilib aytganda, xotiraning segment-sahifali taqsimlanishi xotira boshqaruvi nazariyasi va amaliyotida muhim o'rin tutadi. Uning prinsiplarini tushunish operatsion tizimlar qanday ishlashini va xotira resurslaridan qanday qilib samarali foydalanish mumkinligini anglash uchun zarurdir. Garchi zamonaviy tendensiyalar soddaroq yondashuvlarga qarab o'zgarayotgan bo'lsa-da, segment-sahifali taqsimlashning asosiy g'oyalari va u taklif qilgan yechimlar xotira boshqaruvi sohasidagi bilimlar uchun qimmatlidir.

Foydalanilgan adabiyotlar

1. Tutorialspoint. (n.d.). *Paged Segmentation and Segmented Paging*. Retrieved May 16, 2025, from <https://www.tutorialspoint.com/paged-segmentation-and-segmented-paging>
2. GeeksforGeeks. (2024, December 28). *Difference Between Paging and Segmentation*. Retrieved May 16, 2025, from <https://www.geeksforgeeks.org/difference-between-paging-and-segmentation/>