

**MASSIVLARNI QAYTA ISHLASHNI O'RGANISH. UZILISHLARNI
TASHKIL QILISH**

Sultonmuratova Mashhura Quvonchbek qizi

TATU Urganch filiali talabasi

E-mail: sultonmuratovamashhura@gmail.com

***Annotatsiya.** Ushbu maqolada massivlarni qayta ishlash va uzilishlarni tashkil qilish haqida so'z yuritiladi. Hozirgi kunda mobil qurilmalar zamonaviy hayotimizning ajralmas qismi sanalib, massivlarni qayta ishlashni o'rganish va uzilishlarni tashkil qilish mikroprotessor va assembler tili darsi jarayonida zamonaviy pedagogik texnologiyalardan unumli foydalanish, o'quvchilarning muhandislik sohasiga bo'lgan qiziqishini yanada oshirish va ularning axborot texnologiyalari bo'yicha egallaydigan bilim, ko'nikma va malakalarini yuqori bo'lishini ta'minlaydi.*

***Аннотация.** В этой статье обсуждается обработка массивов и управление прерываниями. В наше время мобильные устройства считаются неотъемлемой частью нашей современной жизни, а обучение обработке массивов и организации прерываний – это эффективное использование современных педагогических технологий в ходе урока микропроцессора и языка ассемблера для дальнейшего повышения интереса учащихся к данной области. инженерного дела, а также повысить их интерес к информационным технологиям, обеспечивает высокий уровень приобретенных знаний, навыков и квалификации.*

***Annotation.** This article discusses array processing and interrupt management. Nowadays, mobile devices are considered an integral part of our modern life, and learning to process arrays and organizing interrupts is an effective use of modern pedagogical technologies in the course of a microprocessor and assembly language lesson, to further increase students' interest in the field of engineering, and to increase their interest in information technology. ensures that*

the acquired knowledge, skills and qualifications are high.

Kalit soʻzlar: assembler tili, massivlar, uzilishlar, mikroprotsessor, oʻzgaruvchilar.

Ключевые слова: язык ассемблера, массивы, прерывания, микропроцессор, переменные.

Key words: assembly language, arrays, interrupts, microprocessor, variables.

Kirish. Assembler tilida yozilgan dasturlar, odatiy dasturlash tillariga nisbatan koʻp yorqinlikka ega boʻlishi sababli, juda tez va samarali ishlaydi. U asosan kompyuter arxitekturasi va mikroprotsessorga bogʻliq boʻlib, dasturlashning koʻp sohalarda qoʻllaniladi, masalan, jadval korrektsiyalash, ilovalarni optimallashtirish va avtomatik boshqaruvda. Assembler tili, kompyuter arxitekturasini tushunishni talab qiladi va yuqori darajadagi boshqarish imkonini beradi. Uning foydali tomoni jismoniy kompyuterlar, sistem korpuslari, avtomatlashtirilgan apparatlar, boshqarish sistemalari va turli muhandislik sohalarda foydalaniladi.

Massiv - bir nechta oʻzgaruvchilarni bir joyda saqlaydigan maʼlumotlar toʻplamidir. Har bir oʻzgaruvchi massivning indeksi orqali identifikatsiya qilinadi. Assembler tilida massivlar oʻzgaruvchilar toʻplami sifatida ishlatiladi. Massivlar, RAM yoki boshqa xotiralarda joylashgan maʼlumotlarni tashkil etish uchun ishlatiladi. Har bir element massivning indeksi orqali murojaat qilinadi.

Assembler tilida massivlar turli xil boʻladi. Quyidagi turli massivlar mavjud:

1. Byte Massivlari: Bu tur massivlar, har bir elementi 1 baytdan iborat boʻlgan maʼlumotlarni saqlayadi. `DB` (Define Byte) direktivasi yordamida belgilanadi.

Misol uchun:

myArray DB 1, 2, 3, 4, 5 ; 5 elementdan iborat byte massivi

2. Word Massivlari: Bu tur massivlar, har bir elementi 2 baytdan iborat boʻlgan maʼlumotlarni saqlayadi. `DW` (Define Word) direktivasi yordamida belgilanadi.

Misol uchun:

myArray DW 100, 200, 300 ; 3 elementdan iborat word massivi

3. Doubleword Massivlari: Bu tur massivlar, har bir elementi 4 baytdan iborat

bo'lgan ma'lumotlarni saqlayadi. `DD` (Define Doubleword) direktivasi yordamida belgilanadi. Misol uchun:

myArray DD 1000, 2000, 3000 ; 3 elementdan iborat doubleword massivi

4. Qator Massivlari: Bu tur massivlar, qatorlardan iborat ma'lumotlarni saqlayadi. `DQ` (Define Qword) direktivasi yordamida belgilanadi. Misol uchun:

myArray DQ "Hello", "World" ; 2 elementdan iborat qator massivi

5. Boshqa Massiv Turlari: Assembler tillarida boshqa massiv turlari ham mavjud bo'lishi mumkin, masalan, Float Massivlari (DF), XMM Massivlari (DQ), SIMD Massivlari va boshqalar. Bu massiv turlari xususiyatlariga va xotiradagi ma'lumotlarning turiga qarab tanlanadi.

Har bir massiv turini belgilash uchun mos direktivalar va sintaksislar ishlatiladi. Mos direktivalar yordamida ma'lumotlar to'plamlari to'g'ridan-to'g'ri xotira joyida va qidiruv uchun mos amallar bajariladi.

Assembler tilida massivni aniqlash uchun quyidagi ko'rsatmalar va direktivalar foydalaniladi:

1. DB (Define Byte): Massiv elementlarini belgilash uchun ishlatiladi. Masalan: `myArray DB 1, 2, 3, 4, 5` massivni aniqlaydi va unda 1, 2, 3, 4, 5 elementlar mavjud.

2. DW (Define Word): 16-bitlik massiv elementlarini belgilash uchun ishlatiladi. Masalan: `myArray DW 100, 200, 300` massivni aniqlaydi va unda 100, 200, 300 elementlar mavjud.

3. DD (Define Doubleword): 32-bitlik massiv elementlarini belgilash uchun ishlatiladi. Masalan: `myArray DD 1000, 2000, 3000` massivni aniqlaydi va unda 1000, 2000, 3000 elementlar mavjud.

Massiv elementlariga murojaat qilish uchun indekslar ishlatiladi. Indekslar 0 dan boshlab massivning uzunligi-1 gacha bo'lgan bo'shliqda bo'lishi mumkin. Elementlarga murojaat uchun indekslar ko'rsatmasi (indexing) va o'zgaruvchi nomi bilan birga ishlatiladi. Masalan: `mov al, myArray[2]` ifodasi massivning 2-indeksli elementini o'qib oladi va uning qiymatini `al` registrga yuklaydi. Massivlar bir nechta ma'lumotlarni bir joyda saqlash, qayta ishlatish va ma'lumotlarni tahlil qilish

imkonini beradi.

Massivlarni qayta ishlash uchun, ularning elementlariga murojaat qilish, qiymatlarini o'zgartirish yoki o'qish, yoki yangi ma'lumotlar bilan to'ldirish mumkin. Assembler tilida massivlarni qayta ishlashning bir necha usullari mavjud, quyidagi misollar hammasini namoyish etish uchun yordam beradi:

1. Elementga murojaat qilish:

mov al, myArray[2] - myArray massivning 2-indeksli elementini o'qib oladi va uning qiymatini al registrga yuklaydi;

2. Elementga qiymat tahrirlash:

mov myArray[3], bl - bl registridagi qiymatni myArray massivning 3-indeksli elementiga saqlaydi;

3. Massivni to'liq qayta ishlash:

lea si, myArray - registrga myArray massivning manzili yuklanadi;

mov cx, length - length o'zgaruvchida massivning uzunligi saqlanadi;

add cx, -1 - uzunlikdan 1 ni ayiradi, chunki indekslar 0 dan boshlanadi;

mov bx, 0 - bx registrga 0 qiymatini yuklaydi;

loop_start:

mov al, [si+bx] - si va bx registrlarni yig'adi;

inc bx - bx registrga 1 qo'shib birinchi elementdan keyingi elementga o'tadi;

loop loop_start - cx registrdagi uzunlikni 1 kamayib borishga takrorlanadi;

Bu misollar massiv elementlariga murojaat qilish, qiymatlarni o'zgartirish va to'liq massivni qayta ishlash uchun yordam beradi.

Massiv elementlarini o'zgartirish uchun quyidagi usullar mavjud:

1. Qiymatni o'zgartirish: Elementning qiymatini o'zgartirish uchun, mos uzilish (indeks yoki element) orqali elementga murojaat qilinadi va yangi qiymat yuklanadi. Misol uchun, quyidagi kod massivning 2-indeksli elementini 10 ga o'zgartiradi:

mov myArray[2], 10 - myArray massivning 2-indeksli elementiga 10 qiymatini yuklash

2. Elementlarni almashtirish: Elementlarni almashtirish (swap) usuli orqali

massivning ikkita elementini o'zgartirish mumkin. Misol uchun, quyidagi kod massivning 1-va 2-indeksli elementlarini almashtiradi:

```
mov al, myArray[1] - 1-indeksli elementni almashtirish uchun alga saqlab  
olamiz;
```

```
xchg al, myArray[2] - 2-indeksli elementni 1-indeksli element bilan  
almashtiramiz;
```

```
mov myArray[1], al - al o'zgaruvchisidagi qiymatni 1-indeksli elementga  
yuklaymiz;
```

3. O'zgaruvchilar orqali o'zgartirish: Massiv elementlarini o'zgaruvchilar orqali o'zgartirish uchun, massivning indeks uzilishini o'zgaruvchi bilan olish va o'zgaruvchi qiymatini o'zgartirish talab etiladi. Misol uchun:

```
mov si, 2 - si o'zgaruvchisiga massivning indeks uzilishini yuklaymiz;  
add myArray[si], 5 - massivning si-indeksli elementiga 5 ni qo'shib  
yuklaymiz.
```

4. Loop yordamida o'zgartirish: Loop (takrorlash) yordamida massivning barcha yoki bir qismi elementlarini o'zgartirish mumkin. Misol uchun, quyidagi kod massivning barcha elementlariga 1 ni qo'shib beradi:

```
lea si, myArray - si registrga massivning manzilini yuklab olamiz;  
mov cx, length - massiv uzunligini cx registrga yuklab olamiz;  
mov bx, 0 - bx registrga 0 qiymatini yuklab olamiz;  
loop_start:  
add byte ptr [si+bx], 1 - massivning bx-indeksli elementiga 1 ni qo'shadi;  
inc bx - bx registrga 1 qo'shib birinchi elementdan keyingi elementga o'tadi;  
loop loop_start - cx registrdagi uzunlikni 1 ga kamayib borishgacha  
takrorlaydi.
```

Uzilishlar tarixi, kompyuter arxitekturasining asosiy bosqichlaridan boshlab, eng so'nggi boshlang'ich kompyuterlarning boshqa turlari va dasturlash tillarining o'sishiga qadar uzaydi. Quyidagi muhim bosqichlar tarixda katta ahamiyatga ega bo'lgan:

1. Boshlang'ich Kompyuterlar va Uzilishlar: Boshlang'ich kompyuterlar, 20-

asr boshida va 20-asr boshidan o'tgan yillarda dasturlash tillarini o'rganish va dasturlash asoslarini rivojlantirish uchun keng foydalanilgan. Bu dori-darmon kompyuterlar, uzilishlarni bajarish uchun elektronik va mexanik komponentlardan foydalanishgan.

2. Uzilishlarning Transistorga O'tishi: 1947 yilida tranzistor yaratilishi, elektronik komponentlarning o'rnatilishi va kompyuter arxitekturasining o'zgarishi uchun katta ahamiyatga ega bo'lgan. Tranzistorlar, elektronik sinalni boshqarish, ma'lumotlarni saqlash va uzilishlarni bajarish jarayonlarida katta o'zgarishlar keltirib chiqardi.

3. Integrlangan Chiplar va Uzilishlar: 1960-yillarda integrlangan tizimlar (IC) yaratildi. IC'larda bir nechta tranzistor, diod va boshqa elektronik komponentlar bir tuzilma ichida integrallangan edi. Bu, kompyuterlarning kichiklashishi, ishga tushirilishi va institutsionalar uchun massalashtirilishi imkonini berdi.

4. Mikroprotessorlar va Uzilishlar: 1970-yillarda mikroprotessorlar yaratilishining boshlang'ici bilan kompyuterlar va uzilishlar sohasida katta o'zgarishlar ro'y berdi. Mikroprotessorlar, bir tuzilma ichidagi barcha kompyuter elementlarini o'z ichiga olgan va dasturlash uchun katta imkoniyatlar yaratdi.

5. So'nggi Rivojlanish: Keyinchalik, uzilishlar va kompyuter arxitekturasini kengayib, kichiklashib, tezlashib, energiya samaradorligi oshib borayotgan. Kvantum kompyuterlar, paralel kompyuterlar, nurol tarmoqlar va boshqa yangi texnologiyalar uzilishlar sohasidagi so'nggi rivojlanishlardan ba'zilaridir.

Uzilishlar, assembler dasturlashda ma'lum elementlarga murojaat qilish uchun ishlatiladigan belgilardir. Uzilishlar, ma'lumotlarni topish, o'zgartirish, o'qish va yozish uchun foydalaniladi. Quyidagi usullar orqali uzilishlar tashkil etiladi:

1. Indeks Uzilishi: Indeks uzilishi, massivning har bir elementiga murojaat qilish uchun ishlatiladi. Indeks uzilishi massivning boshidan boshlab bo'shligacha bo'lgan butun son bo'lishi kerak. Masalan, `[0]` massivning birinchi elementiga, `[1]` massivning ikkinchi elementiga murojaat qilish uchun ishlatiladi.

2. Offset Uzilishi: Offset uzilishi, xotiradagi joyning belgilanishi uchun ishlatiladi. Offset uzilishi massivning boshidan boshlab berilgan elementning

joylashgan xotiradagi manzilini belgilaydi.

3. Base Uzilishi: Base uzilishi, massivlarni yig'ish uchun ishlatiladi. Base uzilishi biror o'zgaruvchi yoki belgilash orqali berilgan xotira manzilini belgilaydi. Massiv elementlariga murojaat qilishda offset uzilishi base uzilishi bilan qo'shib ishlatiladi. Misol uchun, `[bx+di]` base uzilishi bo'lib, `bx` va `di` o'zgaruvchilarni qo'shish orqali massiv elementlariga murojaat qilish uchun ishlatiladi.

4. Scale Uzilishi: Scale uzilishi, massivlarda elementlar orasidagi bo'shliklarni belgilash uchun foydalaniladi. Scale uzilishi massiv elementlarining o'rnini belgilash uchun ishlatilgan bo'ladi. Quyidagi scale uzilishlari mavjud: `1`, `2`, `4`, `8`. Misol uchun `[bx+di*2]` scale uzilishiga misol bo'ladi.

Uzilishlar, dasturlash tillariga qarab o'zgaruvchilar yoki ma'lumotlarni ifodalash uchun ishlatiladi. Massivlar esa o'zgaruvchilarni bir qator elementlarga saqlash uchun ishlatiladi. Massiv, biror turlardagi ma'lumotlarni bir joyda saqlashga imkon beradi. Indeks uzilishi orqali massiv elementlariga murojaat qilish mumkin. Quyidagi misol orqali uzilishlar va massivlarning bir xil dasturda qanday ishlatilishi ko'rsatilgan:

```
# Uzilishlar va massivlar misoli Python dasturida
# Massivni e'lon qilish
my_array = [1, 2, 3, 4, 5]
# Massivning elementiga murojaat qilish
print(my_array[2]) # Natija: 3
# Uzilishlar bilan ma'lumotni o'zgartirish
my_array[1] = 10
print(my_array) # Natija: [1, 10, 3, 4, 5]
```

Ushbu misolda, `my_array` nomli massiv e'lon qilingan. Massivning indeksi orqali `[2]` uzilishi bilan massivning 2-indexli elementiga murojaat qilindi va natijada `3` qaytarildi. Keyin, `[1]` uzilishi orqali massivning 1-indexli elementi `2` dan `10` ga o'zgardi. Natijada massiv `[1, 10, 3, 4, 5]` ga o'zgardi.

Uzilishlarni assembler tilida yozish, dastur avtomatik ravishda bajarish uchun kompyuter tomonidan ko'rib chiqish jarayonida muhim bo'lgan ma'lumotlar va

buyruqlarni belgilashni ta'minlaydi. Assembler tilida yozilgan dasturlar, bajarish jarayonida kompyuter tomonidan aniq ravishda bajariladi. Bu til, bajarish tezligi va samaradorligi uchun muhimdir, chunki assembler tilida yozilgan dasturlar direkt ravishda kompyuterning boshqa tillarga nisbatan yuqori darajada optimallashtirilgan holda ishlaydi. Shu sababli, assembler tilida yozilgan uzilishlar, kompyuter arxitekturasini to'liq nazorat qilish va dasturlarni samarali bajarish uchun katta ahamiyatga ega bo'ladi. Assembler tilida yozilgan dasturlash, xususiy kompyuter arxitekturasiga mos kelgan yuqori darajadagi nazorat va optimallashtirish imkonini beradi.

Xulosa. Mazkur maqolada massivlarni qayta ishlash va uzilishlarni tashkil qilish, shuningdek, ulardan samarali foydalanish haqida so'z yuritiladi. Uzilishlar, assembler tilida dasturlashda katta ahamiyatga ega bo'lgan. Assembler, kompyuterlar uchun eng yaqin dasturlash tillaridan biridir va uzilishlarni bajarish uchun eng yaqin tilga mos keladi. Assembler tilida yozilgan dasturlar, kompyuter arxitekturasining xususiyatlarini to'liq ravishda foydalanish imkonini beradi. Uzilishlarni assembler tilida yozish, dastur avtomatik ravishda bajarish uchun kompyuter tomonidan ko'rib chiqish jarayonida muhim bo'lgan ma'lumotlar va buyruqlarni belgilashni ta'minlaydi.

FOYDALANILGAN ADABIYOTLAR

1. Mikroprotsessor (X.Yu.Abasxonova, M.B.Mirzayeva, S.S.Parsiyev)
2. Gerasimenko V.A. *Zashchita informatsii v avtomatizirovannykh sistemax obrabotki dannykh kn. 1.-M.: Energoatomizdat. -1994.-400s.*
3. Massivlarni tashkil etish (Gulomov S.S.)
4. Stamp Mark. *Information security: principles and practice. USA, 2011.*

Internet resurslari

6. <http://www.google.com>
7. <http://www.it-ebooks.info>
8. www.poe.com