**ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ**

**Мэдээлэл холбооны технологийн сургууль**

**A blue and yellow sign with a gear and drops

Description automatically generated**

**ЛАБОРАТОРИЙН АЖЛЫН**

**ТАЙЛАН**

**Доод түвшиний программчлал (F.NSA233)**

**2023-2024 оны хичээлийн жилийн Хавар**

**Лабораторийн ажлын нэр:** Лабораторийн Ажил №1

**Хичээл заасан багш:** Б.Мөнхбаяр

**Лабораторийн ажил гүйцэтгэсэн:** Оюутан: Б.Түшигт/В221870516/

Б.Мөнхбаяр/В221870521/

|  |  |  |  |
| --- | --- | --- | --- |
| **Авбал зохих нийт оноо:** | **5 оноо** |  |  |
|  | | | |
| **Гүйцэтгэлийн шалгуур** | **Үнэлгээний**  **эзлэх хувь** | **CLOs**  **хамаарал** | **Багшийн**  **үнэлгээ** |
| Хичээлийн ирц/оролцоо | 10% | 4-9 |  |
| Доод түвшиний прораммчлалын симуляци болон багцад дүн шинжилгээ хийх програмыг ашиглах | 10% | 4 |  |
| Лабораторийн заавар, дарааллын дагуу жишээг хийж  гүйцэтгэх | 10% | 5-8 |  |
| Лабораторийн заавар дээр өгөгдсөн даалгаврыг хий гүйцэтгэх | 40% | 5-8 |  |
| Лабораторийн тайлан бичилт, хамгаалалт | 30% | 9 |  |
| Нийт үнэлгээ /хувиар/ | | |  |
| Нийт үнэлгээ /оноогоор/ | | |  |

## Нэгдүгээр лабораторийн ажил Computer architecture

### Лабораторийн ажлын зорилго:

Энэхүү лабораторын ажлаар тооллын систем хооронд хөрвүүлгэ хийхэд суралцана.

### Лабораторийн ажлын суралцахуйн үр дүнгүүд:

Энэ лабораторийн ажлыг гүйцэтгэснээр оюутан дараах чадваруудтай болсон байна.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **д/д** | **Суралцахуйн үр дүнгүүд** | **Суралцахуйн үр дүнг илэрхийлэх үйл үг** | **Суралцахуй н үр дүнгийн түвшин (Блумын)** | **CLOs**  **хамаарал** |
| 1 | (программыг дуусгах) halt program |  |  |  |
| 2 | (тэмдэгт мөрийг хэвлэх) print string |  |  |  |
| 3 | (тэмдэгт мөрийг унших) read string |  |  |  |
| 4 | (бүхэл тоог хэвлэх) print integer |  |  |  |
| 5 | (бүхэл тоог унших) read integer |  |  |  |

### Ашиглах програм хангамж/техник хангамж, бусад хэрэглэгдэхүүнүүд:

Энэхүү лабораторийн ажлыг хийж гүйцэтгэхэд дэвтэр болон үзэг хэрэглэгдэнэ.

### Онолын ойлголтууд:

**2 -тын системээс 10 -тын систем руу хөрвүүлэх**

Харин тоог 2 -тын системээс 10 -тын систем рүү хөрвүүлэхэд энэ аргыг урвуугаар нь хэрэглэнэ.

Жишээлбэл: 10011110110b = 20+21+22+23+24+25+26+27+28+29+210 =1270 . Энд

хоёртын тоог хойноос нь эхлэн харгалзах хоёртын зэрэгтуудийн нийлбэрт задлах ба тухайн байрлалд 0 байвал 0ийг, харин 1 байвал тухайн байрлалд харгалзах хоёртын зэрэгтийг нэмэх замаар харгалзах аравтын тоог гаргана.

2 -тын системээс 16 -тын систем pүү хөрвүүлэх

Хоёрт ба 16-тын системийн сууриуд 16=24 гэсэн холбоотой учир тэдгээрийн хооронд тоог шууд хөрвүүлж болдог.

Ингэхдээ хоёртын тоог араас нь дөрөв дөрвөн оронгоор нь бүлэглэн тухайн хоёртын тоонд харгалзах 16 -тын тоог тавина.

Жишээлбэл: 010011110110 = 0100'1111'0110 = 4'F'6h > 0011'1111'0111b = 3'F'7h

### Суралцахуйн үр дүнг үнэлэх даалгаварууд:

1. Өгөгдсөн аравтын тоог хоёртын тооллын системд хөрвүүл.

1. Өгөгдсөн аравтын тоог хоёртын тооллын системд хөрвүүл.

1. 403 = 2^8 + 2^7 + 2^4 + 2^1 + 2^0 = 110010011
2. 236 = 2^7 + 2^6 + 2^5 + 2^3 + 2^2 = 11101100
3. 111 = 2^6 + 2^5 + 2^3 + 2^2 + 2^1 + 2^0 = 01101111
4. 405 = 2^8 + 2^7 + 2^4 + 2^2 + 2^0 = 110010101
5. 550 = 2^9 + 2^5 + 2^2 + 2^1 = 1000100110
6. 772 = 2^9 + 2^8 + 2^1 + 2^0 = 1100000100
7. 3441 = 2^11 + 2^10 + 2^8 + 2^6 + 2^5 + 2^4 + 2^0 = 110101110001
8. 7020 = 2^12 + 2^11 + 2^9 + 2^8 + 2^6 + 2^5 + 2^3 + 2^2 = 1101101101100
9. 5768 = 2^12 + 2^10 + 2^9 + 2^7 + 2^6 + 2^2 = 1011011011000
10. 2196 = 2^11 + 2^7 + 2^4 + 2^2 = 100010001100

2. Дээр хөрвүүлсэн хоёртын тоонуудаа 16 тын тооллын системд бич.

1. 403 = 110010011 -> 0011 0010 011 -> 0x193
2. 236 = 11101100 -> 1110 1100 -> 0xEC
3. 111 = 01101111 -> 0110 1111 -> 0x6F
4. 405 = 110010101 -> 0110 0101 -> 0x195
5. 550 = 1000100110 -> 0101 1001 0110 -> 0x226
6. 772 = 1100000100 -> 0111 1000 0010 -> 0x304
7. 3441 = 110101110001 -> 0011 0101 1100 0001 -> 0xD71
8. 7020 = 1101101101100 -> 0111 0000 0011 0110 -> 0x1B7C
9. 5768 = 1011011011000 -> 0101 0111 0110 1000 -> 0x1688
10. 2196 = 100010001100 -> 0010 0001 1001 1000 -> 0x891

Top of Form

3. Дараах хоёртын тоонуудыг 10 тын тооллын системд хөрвүүл.

1. 01000010b = 2^6 + 2^1 = 64 + 2 = 66
2. 00010011b = 2^4 + 2^1 + 2^0 = 16 + 2 + 1 = 19
3. 00111001b = 2^5 + 2^4 + 2^3 + 2^0 = 32 + 16 + 8 + 1 = 57
4. 01000110b = 2^6 + 2^3 + 2^2 + 2^1 = 64 + 8 + 4 + 2 = 78
5. 01010001b = 2^6 + 2^4 + 2^0 = 64 + 16 + 1 = 81
6. 10011011b = 2^7 + 2^4 + 2^3 + 2^1 + 2^0 = 128 + 16 + 8 + 2 + 1 = 155
7. 00100101b = 2^5 + 2^2 + 2^0 = 32 + 4 + 1 = 37
8. 01001001b = 2^6 + 2^3 + 2^0 = 64 + 8 + 1 = 73
9. 00011001b = 2^4 + 2^3 + 2^0 = 16 + 8 + 1 = 25
10. 00011110b = 2^4 + 2^3 + 2^2 + 2^1 = 16 + 8 + 4 + 2 = 30
11. 10110001b = 2^7 + 2^6 + 2^4 + 2^0 = 128 + 64 + 16 + 1 = 209
12. 10110010b = 2^7 + 2^6 + 2^4 + 2^1 = 128 + 64 + 16 + 2 = 210
13. 10111110b = 2^7 + 2^6 + 2^5 + 2^4 + 2^1 = 128 + 64 + 32 + 16 + 2 = 242
14. 11010000b = 2^7 + 2^6 + 2^4 = 128 + 64 + 16 = 208
15. 11010010b = 2^7 + 2^6 + 2^4 + 2^1 = 128 + 64 + 16 + 2 = 210
16. 10010110b = 2^7 + 2^4 + 2^2 + 2^1 = 128 + 16 + 4 + 2 = 150
17. 01100101b = 2^6 + 2^5 + 2^2 + 2^0 = 64 + 32 + 4 + 1 = 101
18. 10000001b = 2^7 + 2^0 = 128 + 1 = 129
19. 10110101b = 2^7 + 2^6 + 2^4 + 2^2 + 2^0 = 128 + 64 + 16 + 4 + 1 = 213
20. 11001000b = 2^7 + 2^6 + 2^4 + 2^3 = 128 + 64 + 16 + 8 = 216

4. Дараах 16 тын тооллын системийн тоонуудыг 2 тын тооллын системд хөрвүүл

1. 01Аh = 16^1 + 10 = 16 + 10 = 26
2. 03Ch = 16^2 + 12\*16^1 = 256 + 192 = 448
3. 043h = 16^2 + 3 = 256 + 3 = 259
4. 03Bh = 16^2 + 11 = 256 + 11 = 267
5. 0F7h = 15\*16^2 + 7 = 3840 + 7 = 3847
6. 020h = 2\*16^1 = 32
7. 0BBh = 11\*16^2 + 11 = 2816 + 11 = 2827
8. 051h = 5\*16^1 + 1 = 80 + 1 = 81
9. 07Bh = 7\*16^1 + 11 = 112 + 11 = 123
10. 077h = 7\*16^1 + 7 = 112 + 7 = 119
11. 097h = 9\*16^1 + 7 = 144 + 7 = 151
12. 030h = 3\*16^1 = 48
13. 0ECh = 14\*16^2 + 12 = 3584 + 12 = 3596
14. 0E5h = 14\*16^2 + 5 = 3584 + 5 = 3589
15. 0FDh = 15\*16^2 + 13 = 3840 + 13 = 3853
16. 0B3h = 11\*16^2 + 3 = 2816 + 3 = 2819
17. 0ADh = 10\*16^2 + 13 = 2560 + 13 = 2573
18. 03Eh = 3\*16^2 + 14 = 768 + 14 = 782
19. 09Ch = 9\*16^2 + 12 = 2304 + 12 = 2316
20. 08h = 8
21. Дараах үйлдлүүдийн 10 тын систем дэх бичиглэлийг 2 тын систем рүү хөрвүүлээд 2 тын системд бод.

1. 29 + 53 = 00011101+00110101=01010010

2. 09 + 07 = 00001001+00000111=00010000

3. 10 + 07 = 00001010+00000111=00010001

1. 21 + 62 = 00010101+00111110=1010011
2. 19 + 84 = 00010011+01010100=01100111
3. Өгөгдсөн хоёртын х , у тоонууд дээр дараах логик үйлдлүүдийг гүйцэтгэ.
   1. z = x OR y

0101 OR 0010 = 0111, 0011 OR 1001 = 1011, 0101 OR 1010 = 1111, 1110 OR

0011 = 1111, 1010 OR 1110 = 1110

* 1. z = x AND y

0101 AND 0010 = 0000, 0011 AND 1001 = 0001, 0101 AND 1010 = 0000, 1110

AND 0011 = 0010, 1010 AND 1110 = 1010

* 1. z = x XOR y

0101 XOR 0010 = 0111, 0011 XOR 1001 = 1110, 0101 XOR 1010 = 1111, 1110 XOR

0011 =1101, 1010 XOR 1110 = 0100

* 1. z = x NOR y

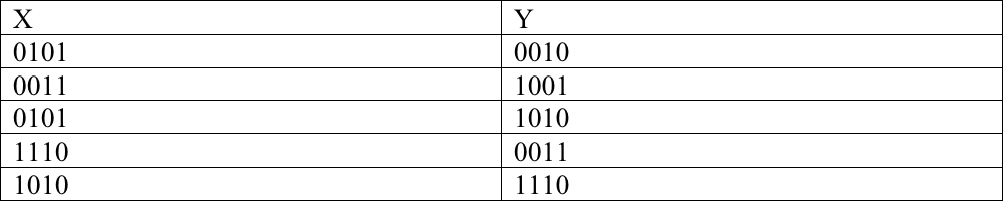
0101 NOR 0010 = 1000, 0011 NOR 1001 = 0100, 0101 NOR 1010 = 0000, 1110

NOR 0011 = 0000, 1010 NOR 1110 = 0001

* 1. z = x NAND y

0101 NAND 0010 = 1111, 0011 NAND 1001 = 1110, 0101 NAND 1010 = 1111,

1110 NAND 0011 = 1101, 1010 NAND 1110 = 0101



**Ашиглсан материал:** Лекц-1 материал Лабораторийн ажил 1

**ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ**

**Мэдээлэл холбооны технологийн сургууль**

**A blue and yellow sign with a gear and drops

Description automatically generated**

**ЛАБОРАТОРИЙН АЖЛЫН**

**ТАЙЛАН**

**Доод түвшиний программчлал (F.NSA233)**

**2023-2024 оны хичээлийн жилийн Хавар**

**Лабораторийн ажлын нэр:** Лабораторийн Ажил №2

**Хичээл заасан багш:** Б.Мөнхбаяр

**Лабораторийн ажил гүйцэтгэсэн:** Оюутан: Б.Түшигт/В221870516/

Б.Мөнхбаяр/В221870521/

|  |  |  |  |
| --- | --- | --- | --- |
| **Авбал зохих нийт оноо:** | **5 оноо** |  |  |
|  | | | |
| **Гүйцэтгэлийн шалгуур** | **Үнэлгээний**  **эзлэх хувь** | **CLOs**  **хамаарал** | **Багшийн**  **үнэлгээ** |
| Хичээлийн ирц/оролцоо | 10% | 4-9 |  |
| Доод түвшиний прораммчлалын симуляци болон багцад дүн шинжилгээ хийх програмыг ашиглах | 10% | 4 |  |
| Лабораторийн заавар, дарааллын дагуу жишээг хийж  гүйцэтгэх | 10% | 5-8 |  |
| Лабораторийн заавар дээр өгөгдсөн даалгаврыг хий гүйцэтгэх | 40% | 5-8 |  |
| Лабораторийн тайлан бичилт, хамгаалалт | 30% | 9 |  |
| Нийт үнэлгээ /хувиар/ | | |  |
| Нийт үнэлгээ /оноогоор/ | | |  |

## Хоёрдугаар лабораторийн ажил

### Лабораторийн ажлын зорилго:

Энэхүү лабораторийн ажлаар SPIM программын анхан шатны мэдэгдэхүүнтэй болно.

### Лабораторийн ажлын суралцахуйн үр дүнгүүд:

Энэ лабораторийн ажлыг гүйцэтгэснээр оюутан дараах чадваруудтай болсон байна.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **д/д** | **Суралцахуйн үр дүнгүүд** | **Суралцахуйн үр дүнг илэрхийлэх үйл үг** | **Суралцахуй н үр дүнгийн түвшин (Блумын)** | **CLOs**  **хамаарал** |
| 1 | (программыг дуусгах) halt program |  |  |  |
| 2 | (тэмдэгт мөрийг хэвлэх) print string |  |  |  |
| 3 | (тэмдэгт мөрийг унших) read string |  |  |  |
| 4 | (бүхэл тоог хэвлэх) print integer |  |  |  |
| 5 | (бүхэл тоог унших) read integer |  |  |  |

### Ашиглах програм хангамж/техник хангамж, бусад хэрэглэгдэхүүнүүд:

QTSpim программ хангамж болон Notepad хэрэглэгдэнэ.

### Онолын ойлголтууд:

**SPIM програмыт эхлүүлэх**

MIPS процессор нь 32 битийн, 32 ширхэг ерөнхий зориулалтын регистртэй. Эдгээр нь дотроо тодорхой регистрүүдэд хуваагддаг. Эхний жишээ болох SPIM програм нь бүхэл тоог (integer) дүрслэхдээ 2 регистрт битүүдийн төлвөөр дүрслэх ба дараа нь тэр 2 - ыгоо нэмэх болно. SPIM програмыт Start товчиндээр даран ажлуулна

### Шууд операнд

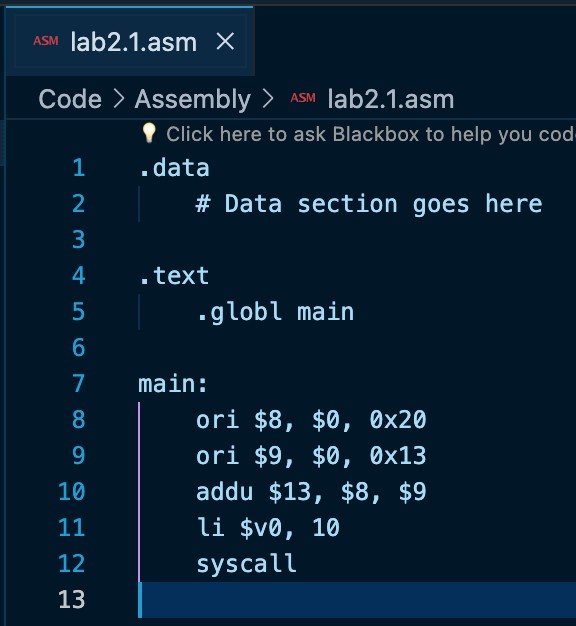
Машины хэлдээрх командын 32 битийн 16 бит нь нэг операндад хэрэглегддэг. Командын энд кодлогдсон нэг операндыг шууд операнд гэж нэрлэдэг. Хаяг машин код ассемблер код 0x00400000 0x34080002 ori $8,$0,0x02 Энэ командын сүүлийн 16 бит нь 0х0002 гэсэн операндыг илтгэж байна (ассемблер хэлдээрх команд доэр 0х2). Мөн ALU, регистр $0 болон шууд операнд 0×0002 - ын хооронд логик OR үйлдэл гүйцэтгэхийг хэлж байна. Үр дүн нь регистр $8 - д орох болно.

### Суралцахуйн үр дүнг үнэлэх даалгаварууд:

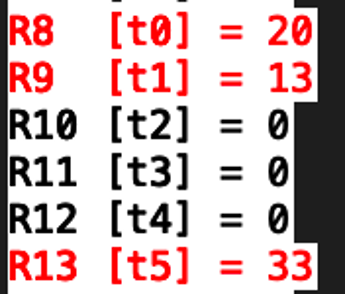
1. р даалгавар:

20 + 13 илэрхийллийг ассемблер хэл дээр бич. Үр дүнг $13 регистрт хадгал.

#### Code-ийн хэсэг:



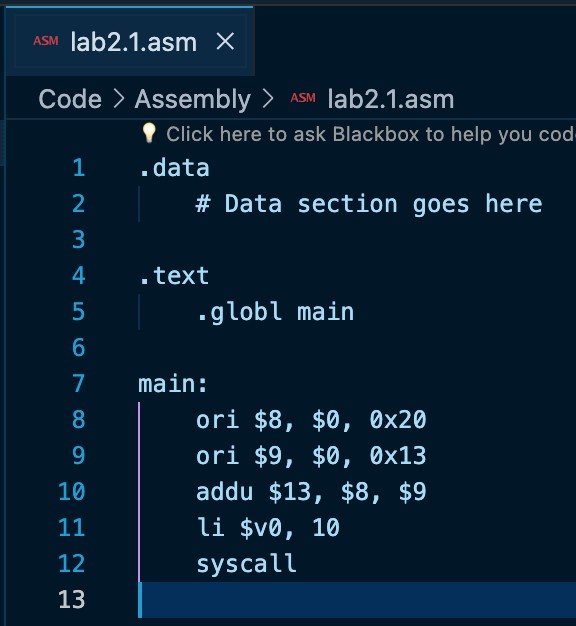
**Үр дүн:**



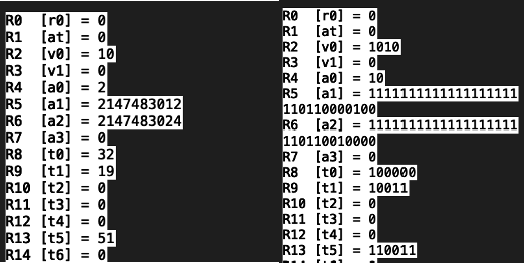
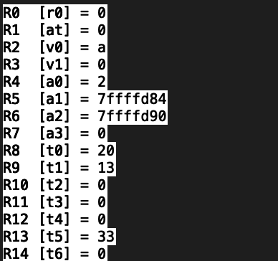
1. р даалгавар:

Гарсан үр дүнг хоёрт, аравт, арван зургаатын тооллын системд тус тус илэрхийл.

**Code-ийн хэсэг:**



**Үр дүн:**



**Hexadecimal Decimal Binary**

1. р даалгавар:

Дараах буулийн функцыг гуйцэтгэх программ бич. y=x1&x2 | x2&x3, y=x1 | x2&x3|×1&x3, y=x1^×3 | ~ (x2|x3 ).

x1= 0xbf x2=0x3a x3=0x74

### Ашиглсан материал:

Лекц-2 материал

Лабораторийн ажил 2 материал

**ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ**

**Мэдээлэл холбооны технологийн сургууль**

**A blue and yellow sign with a gear and drops

Description automatically generated**

**ЛАБОРАТОРИЙН АЖЛЫН**

**ТАЙЛАН**

**Доод түвшиний программчлал (F.NSA233)**

**2023-2024 оны хичээлийн жилийн Хавар**

**Лабораторийн ажлын нэр:** Лабораторийн Ажил №3

**Хичээл заасан багш:** Б.Мөнхбаяр

**Лабораторийн ажил гүйцэтгэсэн:** Оюутан: Б.Түшигт/В221870516/

Б.Мөнхбаяр/В221870521/

|  |  |  |  |
| --- | --- | --- | --- |
| **Авбал зохих нийт оноо:** | **5 оноо** | | |
|  | | | |
| **Гүйцэтгэлийн шалгуур** | **Үнэлгээний эзлэх хувь** | **CLOs**  **хамаарал** | **Багшийн үнэлгээ** |
| Хичээлийн ирц/оролцоо | 10% | 4-9 |  |
| Доод түвшиний прораммчлалын симуляци болон багцад дүн шинжилгээ хийх програмыг ашиглах | 10% | 4 |  |
| Лабораторийн заавар, дарааллын дагуу жишээг хийж гүйцэтгэх | 10% | 5-8 |  |
| Лабораторийн заавар дээр өгөгдсөн даалгаврыг хий гүйцэтгэх | 40% | 5-8 |  |
| Лабораторийн тайлан бичилт, хамгаалалт | 30% | 9 |  |
| Нийт үнэлгээ /хувиар/ | | |  |
| Нийт үнэлгээ /оноогоор/ | | |  |

## Гуравдугаар лабораторийн ажил

### Лабораторийн ажлын зорилго:

Энэ лабораторийн ажлаар MIPS процессорын 32 битийн бүхэл тоон дээр хийгдэх арифметик нэмэх болон хасах үйлдлийн талаар суралцана.

### Лабораторийн ажлын суралцахуйн үр дүнгүүд:

Энэ лабораторийн ажлыг гүйцэтгэснээр оюутан дараах чадваруудтай болсон байна.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **д/д** | **Суралцахуйн үр дүнгүүд** | **Суралцахуйн үр дүнг илэрхийлэх үйл үг** | **Суралцахуй н үр дүнгийн түвшин (Блумын)** | **CLOs**  **хамаарал** |
| 1 | Тэмдэггүй тоо ба хоёртын гүйцээлт, орон халилт |  |  |  |
| 2 | ADD болон ADDU команд |  |  |  |
| 3 | Тэмдгийн өргөтгөл |  |  |  |
| 4 | ADDI болон ADDIU команд |  |  |  |
| 5 | SUB болон SUBU команд |  |  |  |
| 6 | ADDI командыг хэрэглэх сөрөг бүхэл тоог регистрт ачаалах |  |  |  |

### Ашиглах програм хангамж/техник хангамж, бусад хэрэглэгдэхүүнүүд:

Notepad болон QTSpim программ хангамжууд ашиглагдана.

### Онолын ойлголтууд:

**Тэмдэггүй тоо ба орон халилт.**

Тэмдэгтүй хоёртын ( unsigned binary ) 8 бит бухэл тоо нь :

28 буюу 256 байна.

Энд тэмдогтуй хоёртын 4 бит операнд доэр орон халилт ууссэнийг харуулж байна. 1111

0111

1001 0000 carry out ( шилжиж гарсан бит) Хэрэв CARRY OUT=1 бол орон халилт ( overflow ) ууссэн.

Харин CARRY OUT=0 бол орон халилт уусээгуй .

### ADDU команд

ADDU команд нь хоёртын нэмэж уйлдлийн алгоритмаар механикаар бодолт хийж 32 битийн 2 операндаас 32 битийн ур дунг боловсруулдаг. addu d, s, t # регистр &<s+t # халилт хамаарахгуй Харин ADD команд нь халилтын уед хоёртын гуйцээлтийг ашигладаг trap - аас шалтгаалдаг. Бусад тохиолдолд ADDU командтай ижил. Тгар нь энгийн машин цикл дох тасалдал юм. Энгийн компьютерийн системд тасалдлын ур дунд удирдлага эргээд уйлдлийн системруу шилждог. add d,s,t # регистр d<s+t # халилтыг тооцно

### ADDIU команд

ADDIU команд нь 16 бит шууд операндыг агуулдаг. ALU командыг гуйцэттэх уед шууд операнд нь тэмдэтэй 32 битэд орготгегдсон байдаг. Нэмэх уйлдлийн уед хоёртын гуйцээлт дэх халилт уусвэл хэрэгсдэггуй. addiu d, s, const # регистр d <s+ const # const нь 16 бит хоёртын гуйцээлт Харин ADDI командын хувьд нэмэх уйлдлийн уед халилтын тасалдал уусдэг. addi d, s, const # регистр d <st const # const нь 16 бит хоёртын гуйцээлт

### Арифметик баруун шилжуулэх

Логик баруун шилжуулэх уйлдэл нь серог бухэл тоог хуваахад хэрэглогддэттуй. Учир нь логик баруун шилжуулэх уйлдэлд ахлах оронд 0 бичигддэг. Эно нь зарим уед тустай, харин ахлах орон тэмдгийн битийг илэрхийлж байхад серег бухэл тоог хуваахад тохиромжтуй.

Харин арифметик баруун шилжуулэлт нь тэмдгийн битийг хаяхгуй эргэж байрлалд нь хуулж тавьдаг.

### Суралцахуйн үр дүнг үнэлэх даалгаварууд:

#### р даалгавар:

0x4 + 0x6=?

#### Code болон Үр дүнгийн хэсэг:

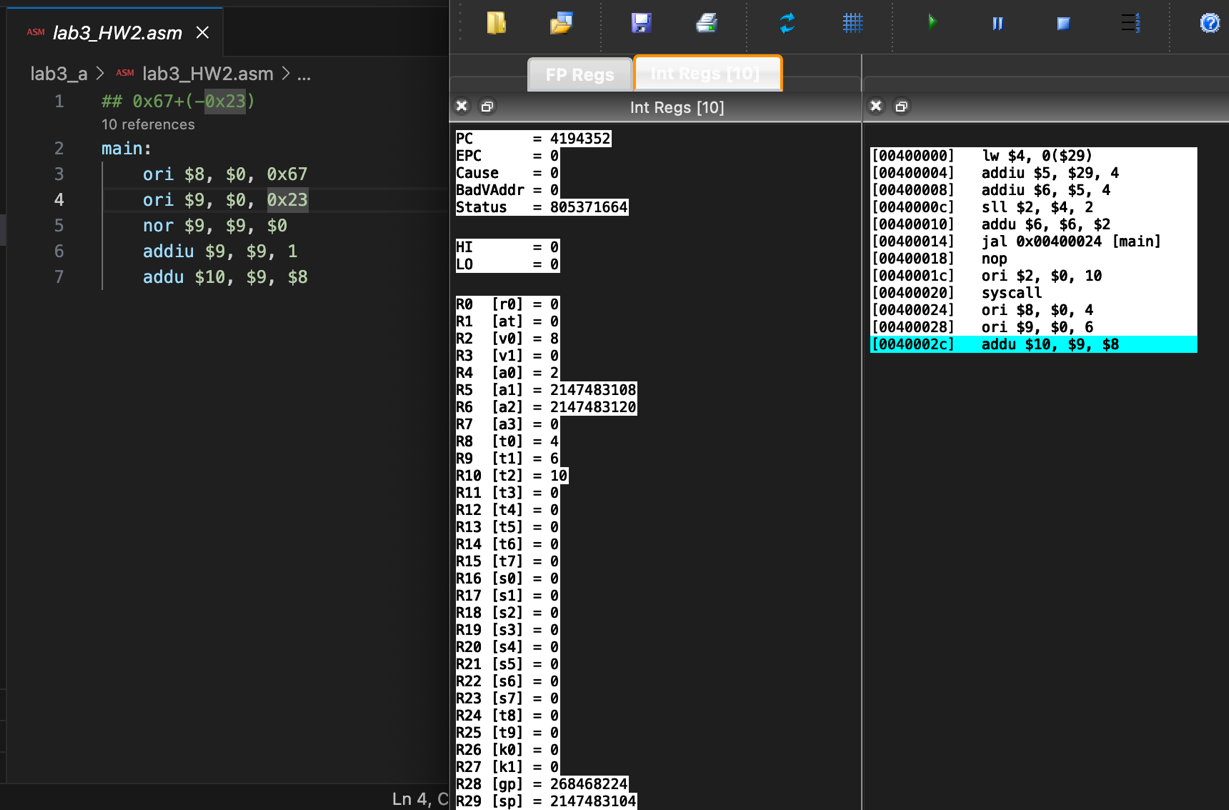
**A screenshot of a computer

Description automatically generated**

1. **р даалгавар:**

0x67 + (-0×23) = ?

#### Code болон Үр дүнгийн хэсэг:

****

1. **р даалгавар:**

0x78 - 0x13 = ?

#### Code болон Үр дүнгийн хэсэг:

**A screenshot of a computer

Description automatically generated**

1. Орон халилт гэж юу вэ?

Арифметик үйлдлийн үр дүн үр дүнд хуваарилагдсан битийн тоогоор илэрхийлэгдэх мужаас хэтэрсэн тохиолдолд халилт үүсдэг.

1. Тэмдгийн бит гэж юу вэ?

Бүхэл тооны хоёртын нэмэлтийн зүүн талын битийг тэмдгийн бит гэнэ.

+5-ийн хувьд: 0x00000005

-5-ын хувьд: 0xffffffff

1. Unsigned гэж юу гэсэн үг вэ?

unsigned гэдэг нь тооны тэмдгийг харгалздаггүй өгөгдлийн төрөл эсвэл үйлдлийг хэлнэ.

1. Хоёртын гүйцээлт гэж юу вэ?

Хоёртын гүйцээлт гэдэг нь өгөгдлийн битүүдийн үгүйсгэл дээр нэгийг нэмэхэд гарах үр дүн юм.

1. Тэмдгийн өргөтгөл гэж юу вэ?

ALU н оролцоотой операндын 32 бит хүртлэх өргөтгөл нь тэмдгийн өргөтгөл юм.

1. Нэмэх үйлдэл ашиглан хасах үйлдэлийг хийж болох уу?

Боломжтой. Хэрэв нэмж байгаа 2 регистрийн аль нэгэнд нь сөрөг утга байвал нэмэх үйлдлээр хасах үйлдлийг хийж болно.

### Ашиглсан материал:

Лекц-3 материал

Лабораторийн ажил 3 материал

**ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ**

**Мэдээлэл холбооны технологийн сургууль**

**A blue and yellow sign with a gear and drops

Description automatically generated**

**ЛАБОРАТОРИЙН АЖЛЫН**

**ТАЙЛАН**

**Доод түвшиний программчлал (F.NSA233)**

**2023-2024 оны хичээлийн жилийн Хавар**

**Лабораторийн ажлын нэр:** Лабораторийн Ажил №4

**Хичээл заасан багш:** Б.Мөнхбаяр

**Лабораторийн ажил гүйцэтгэсэн:** Оюутан: Б.Түшигт/В221870516/

Б.Мөнхбаяр/В221870521/

|  |  |  |  |
| --- | --- | --- | --- |
| **Авбал зохих нийт оноо:** | **5 оноо** | | |
|  | | | |
| **Гүйцэтгэлийн шалгуур** | **Үнэлгээний эзлэх хувь** | **CLOs хамаарал** | **Багшийн үнэлгээ** |
| Хичээлийн ирц/оролцоо | 10% | 4-9 |  |
| Доод түвшиний прораммчлалын симуляци болон багцад дүн шинжилгээ хийх програмыг ашиглах | 10% | 4 |  |
| Лабораторийн заавар, дарааллын дагуу жишээг хийж гүйцэтгэх | 10% | 5-8 |  |
| Лабораторийн заавар дээр өгөгдсөн даалгаврыг хий гүйцэтгэх | 40% | 5-8 |  |
| Лабораторийн тайлан бичилт, хамгаалалт | 30% | 9 |  |
| Нийт үнэлгээ /хувиар/ | | |  |
| Нийт үнэлгээ /оноогоор/ | | |  |

**УЛААНБААТАР ХОТ**

**2024 ОН**

|  |  |  |
| --- | --- | --- |
| **ГАРЧИГ** | | |
|  | | |
| Лабораторийн хичээлийн хөтөлбөр | | 3 |
| Лабораторт тавигдах шаардлага, мөрдөх журам ба аюулгүй ажиллагааны заавар | | 5 |
| Лабораторийн ажлууд | | 7 |
| Ажлын дугаар | Ажлын нэр |  |
| 1 | Тооллын систем | 7 |
| 2 | Лабораторийн ажил №4 |  |
| 3 | Лабораторийн ажил №5 |  |

**Тайлбар:** Лабораторийн хичээлийн хөтөлбөрт туссан мэдээллүүд нь хичээлийн хөтөлбөрийн мэдээлэлтэй ижил бөгөөд зөвхөн дараах агуулгыг нэмж оруулсан болно.

**Гүйцэтгэлийн шалгуур:** Лабораторийн хичээлийн гүйцэтгэлийг үнэлэх шалгуурууд(таны хичээлийн хувьд ялгаатай байж болно) бөгөөд лабораторийн ажил бүрийг эдгээр шалгуурын дагуу үнэлж нэгтгэн CLO бүрийн  гүйцэтгэлийн түвшинг тогтооно. Энд тодорхойлсон шалгууруудын дагуу тодорхойлсон  үнэлгээний хүснэгтийг лабораторийн тайлангийн хавтас дээр оруулсан ба энэ дагуу багш оюутан бүрд үнэлгээ өгч баталгаажуулсан байна.

**Дөрөв дүгээр лабораторийн ажил**

**Регистр, санах ойн зохион байгуулалт**

**Лабораторийн ажлын зорилго:**

Энэхүү лаборторийн ажлаар регистрээс санах ой руу, санах ойгоос регистер лүү өгөгдөл дамжуулах аргын талаар судлах болно.

**Лабораторийн ажлын суралцахуйн үр дүнгүүд:**

Энэ лабораторийн ажлыг гүйцэтгэснээр оюутан дараах чадваруудтай болсон байна.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| д/д | Суралцахуйн үр дүнгүүд | Суралцахуйн үр дүнг илэрхийлэх үйл үг | Суралцахуйн үр дүнгийн түвшин  (Блумын) | CLOs хамаарал |
| 1 | Санах ойн зохион зохион байгуулалт |  |  |  |
| 2 | Регистр |  |  |  |

**Ашиглах програм хангамж/техник хангамж, бусад хэрэглэгдэхүүнүүд:**

PCspim, notepad

**Онолын ойлголтууд:**

Санах ойн зохион байгуулалт, хаягжилт болон түүн дээрх үйлдлүүдийн талаарх ойлголт

**Ажил гүйцэтгэх дараалал:**

**Жишээ програм:**

## endian.asm

## copy $9 to memory in big-endian order

## $8 ----first byte of the tape buffer

## $9 ----4-byte integer

.text

.globl main

main:

lui $9,0x1234 #put data in $9

ori $9,0x5678 # R9  (t1) = 12345678

lui $8,0x1000 # $8 is base register R8  (t0) = 10000000

sb  $9,3($8)  # least significiant byte

srl $9,$9,8   # move next byte to low order

sb $9,2($8)   # bits 8-15 R9  (t1) = 00123456

srl $9,$9,8   # move next byte to low order

sb $9,2($8)  # bits 16-23 R9  (t1) = 00001234

srl $9,$9,8   # move next byte to low order

sb $9,2($8)  # most signifiviant byte R9  (t1) = 00000012

.data

tape: # base register points here

.space 1024 #tape buffer (1K bytes)

##End of file

**Даалгавар:**

.text

.globl main

main:

ori $1,$0,17 #$1-д тогтмол утга хадгална.

lui $2,0x1000 #2-д бааз хаяг хадгалан тогтмол 0х100000000 хаяг байна.

lb $3, 0($2) # санах ойд зарласан х-ийн утгыг $3-д ачааллана.

sll $0,$0,0

sll $0,$0,0 #delay

lb $4, 1($2) #санах ойд зарласан y-ийн утгыг $4-д ачааллана.

sll $0,$0,0

sll $0,$0,0 #delay

multu $3, $4

mflo $5

multu $5, $1

mflo $5 #17xy=$5

ori $1,$0,12

multu $3, $1

mflo $6 #12x=$6

ori $1,$0,6

multu $4, $1

mflo $7 #6y=$7

subu $8,$5,$6 #17xy-12x=$8

subu $8,$8,$7 #17xy-12x-6y=$8

addiu $8,$8,12 #17xy-12x-6y+12=$8

sw $8,4($2) #үр дүнг санах ойн 0х100000004 хаяганд хадгалах

.data #Өгөгдлийн санах ой энэ хэсэгт өгөгдөл зарлана.

x:.byte 0xb

y:.byte 0x5b

**Үр дүн:**

**A table with numbers and symbols

Description automatically generated with medium confidence  
Шалгах асуулт:**1.Ямар үйлдэл нь санах ойгоос ерөнхий зориулалтын регистр рүү өгөгдлийг хуулдаг вэ?

Санах ойгоос ерөнхий зориулалтын регистр рүү өгөгдлийг хуулахдаа **lw (Load Word)** коммандыг ашигладаг. Энэ комманд нь санах ойн хаягаас авсан хаягийн утгыг ерөнхий зориулалтын регистр рүү оруулдаг.

2. Ямар үйлдэл нь ерөнхий зориулалтын регистрээс санах ойгоос рүү өгөгдлийг хуулдаг вэ?

Ерөнхий зориулалтын регистрээс санах ойгоос өгөгдлийг хуулахдаа **sw (Store Word)** коммандыг ашигладаг.

3. Big endian ямар үүрэгтэй вэ?

Big endian буюу том төгсгөлтэй  байтын дараалал нь мэдээллийн хамгийн чухал хэсгийг хадгалж үлдэн бага санах ойд хадгалахад гол үүрэгтэй.

5. Offset хаяг гэж ямар хаягийг хэлдэг вэ?

Үндсэн хаягийг өргөтгөн тодорхой урттай болгох зорилгоор нэмсэн хаягийг ийнхүү нэрийддэг.

4. Lui команд яаж ажилладаг вэ?

16 битийн шууд операндыг тухайн регистрийн ахлах битэд олгох замаар ажилладаг.

**Ашиглах материал:**

Лекц 4 материал

**ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ**

**Мэдээлэл холбооны технологийн сургууль**

**A blue and yellow sign with a gear and drops

Description automatically generated**

**ЛАБОРАТОРИЙН АЖЛЫН**

**ТАЙЛАН**

**Доод түвшиний программчлал (F.NSA233)**

**2023-2024 оны хичээлийн жилийн Хавар**

**Лабораторийн ажлын нэр:** Лабораторийн Ажил №5

**Хичээл заасан багш:** Б.Мөнхбаяр

**Лабораторийн ажил гүйцэтгэсэн:** Оюутан: Б.Түшигт/В221870516/

Б.Мөнхбаяр/В221870521/

|  |  |  |  |
| --- | --- | --- | --- |
| **Авбал зохих нийт оноо:** | **5 оноо** | | |
|  | | | |
| **Гүйцэтгэлийн шалгуур** | **Үнэлгээний эзлэх хувь** | **CLOs хамаарал** | **Багшийн үнэлгээ** |
| Хичээлийн ирц/оролцоо | 10% | 4-9 |  |
| Доод түвшиний прораммчлалын симуляци болон багцад дүн шинжилгээ хийх програмыг ашиглах | 10% | 4 |  |
| Лабораторийн заавар, дарааллын дагуу жишээг хийж гүйцэтгэх | 10% | 5-8 |  |
| Лабораторийн заавар дээр өгөгдсөн даалгаврыг хий гүйцэтгэх | 40% | 5-8 |  |
| Лабораторийн тайлан бичилт, хамгаалалт | 30% | 9 |  |
| Нийт үнэлгээ /хувиар/ | | |  |
| Нийт үнэлгээ /оноогоор/ | | |  |

**УЛААНБААТАР ХОТ**

**2024 ОН**

**Тавдугаар лабораторийн ажил**

**Нөхцөлт үйлдэл**

**Лабораторийн ажлын зорилго:**

Энэхүү лаборторийн ажлаар үсрэлт нөхцөлт үсрэлтийн талаар ойлголт авч хялбар бодлого хэрхэн бодогддогийг судлах болно.

**Ашиглах програм хангамж/техник хангамж, бусад хэрэглэгдэхүүнүүд:**

-PCSpim програм хангамж

**Онолын ойлголтууд:**

Үсрэлт нөхцөлт үсрэлтийн үндсэн ойлголт

**Ажил гүйцэтгэх дараалал:**

**Даалгавар 1:**

**A black text on a white background

Description automatically generated**

**Код:**

.text

.globl main

main:

    # digits массивын хаягийг ачаална

    la $a0, digits

    # digits массивын элементүүдийн тоог тоолно

    li $a1, 20

    # sorting функц руу шууд явуулна

    jal sorting

    # digits массивын хаягийг $t1 рүү хадгална

    la $t1, digits

    # counter-ийг тооцоолох, эхлэл нь 0

    li $t0, 0

print:

    beq $t0, $a1, exit

    # digits массивын элементийг уншиж $a0 рүү хадгалав

    lw $a0, 0($t1)

    # $a0 рүү хадгалсан утгыг хэвлэх

    li $v0, 1

    syscall

    # line орон нь хэвлэгдэх

    li $v0, 4

    la $a0, line

    syscall

    # digits массивын хаягийг дараах элементийн хаяг рүү шилжүүлнэ

    addi $t1, $t1, 4

    addi $t0, $t0, 1

    # print функц руу үсрэнэ

    j print

exit:

    # програмаас гарах

    li $v0, 10

    syscall

sorting:

    # $t2-д 1 утга олгоно

    li $t2, 1

    # $t3-ийг (n - 1)-р санах, эндээс n нь digits массивын элементийн тоо

    sub $t3, $a1, 1

loop2:

    # $t2 0 бол  loop2\_end рүү шилжих

    beqz $t2, loop2\_end

    # $a0 руу хадгалах

    move $t4, $a0

    # $t2 нь 0-ээр санах

    li $t2, 0

    # $t0 нь 0-ээр санах

    li $t0, 0

loop1:

    # $t0 нь $t3-с их байна гэж loop2 руу шилжүүлнэ

    bge $t0, $t3, loop2

    # $t4-ийн 0 байна

    lw $t5, 0($t4)

    # $t4-ийн 4 байна

    lw $t6, 4($t4)

    # $t5 нь $t6-с бага байвал do\_change руу шилжүүлнэ

    blt $t5, $t6, do\_change

    # skip\_change руу шилжүүлнэ

    j skip\_change

do\_change:

    # $t6 руу $t5 хадгална

    sw $t6, 0($t4)

    # $t5 руу $t6 хадгална

    sw $t5, 4($t4)

    # $t2-ийг 1 болгоно

    li $t2, 1

skip\_change:

    # $t4 г 4 нэмэгдүүлнэ

    addi $t4, $t4, 4

    # $t0 руу 1 нэмэгдүүлнэ

    addi $t0, $t0, 1

    # loop1 руу шилжүүлнэ

    j loop1

loop2\_end:

    # $ra руу шилжинэ

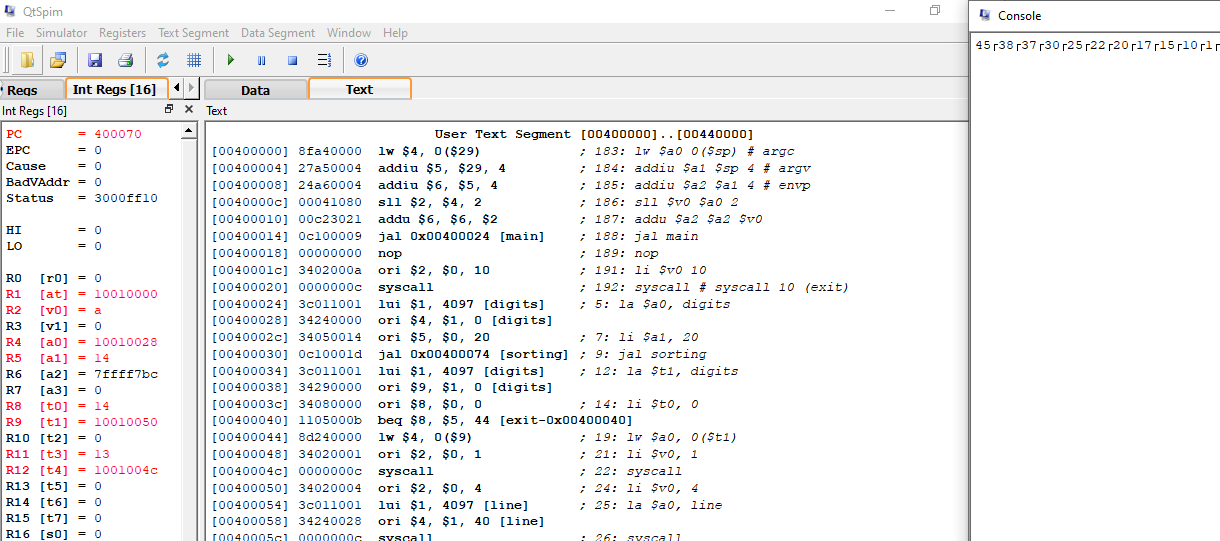
    jr $ra

.data

digits: .word 1, 17, 22, 25, 20, 37, 45, 38, 15, 30

line: .asciiz "\n"

**Үр дүн:**



**Суралцахуйн үр дүнг үнэлэх даалгаварууд:**

1. Машин циклийн гурван алхам юу вэ?

Машин циклийн гурван алхам нь Fetch, PC утга өсгөх,болон Execute хийх гэсэн гурван үйлдлийг илэрхийлэх байдлаар компьютерийн процессорын ажлыг тодорхойлдог.

1. Jump командыг дагаж явдаг дөрвөн байтыг юу гэдэг вэ?

High-Order гэдэг.

1. Pipeline гэж юу вэ?

Машин ажиллаж байх үед нэгэн зэрэг олон үйлдэл хийх шаардлагатай болох үед тулгардаг бэрхшээл бөгөөд үүнээс Delay ашиглаж сэргийлдэг.

1. Jump командын 32 бит хаяг яаж үүсдэг вэ?

Хаягийн зүүн 2 шилжилт хийж ахлах 4 орон дээр инверс авснаар 32 бит хаяг үүсдэг.

1. Branch delay slot-ийг яагаад хийдэг вэ?

Машиний тоолуур ажиллаж байх үед нэгэн зэрэг олон үйлдэл хийх шаардлагатай болдог ба машин нэгэн зэрэг үйлдэл хийдэггүй. Иймээс үйлдлийн дарааллыг зохион байгуулахын тулд Branch delay slot хийдэг.

**Ашиглах материал:**

Лабортори 5 ын материал.

**ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ**

**Мэдээлэл холбооны технологийн сургууль**

**A blue and yellow sign with a gear and drops

Description automatically generated**

**ЛАБОРАТОРИЙН АЖЛЫН**

**ТАЙЛАН**

**Доод түвшиний программчлал (F.NSA233)**

**2023-2024 оны хичээлийн жилийн Хавар**

**Лабораторийн ажлын нэр:** Лабораторийн Ажил №6

**Хичээл заасан багш:** Б.Мөнхбаяр

**Лабораторийн ажил гүйцэтгэсэн:** Оюутан: Б.Түшигт/В221870516/

Б.Мөнхбаяр/В221870521/

|  |  |  |  |
| --- | --- | --- | --- |
| **Авбал зохих нийт оноо: 5 оноо** | | | |
| **a** | | | |
| **Гүйцэтгэлийн шалгуур** | **Үнэлгээний**  **эзлэх хувь** | **CLOs**  **хамаарал** | **Багшийн**  **үнэлгээ** |
| Хичээлийн ирц/оролцоо | 10% | 4-9 |  |
| Доод түвшиний прораммчлалын симуляци болон  багцад дүн шинжилгээ хийх програмыг ашиглах | 10% | 4 |  |
| Лабораторийн заавар, дарааллын дагуу жишээг хийж  гүйцэтгэх | 10% | 5-8 |  |
| Лабораторийн заавар дээр өгөгдсөн даалгаврыг хий  гүйцэтгэх | 40% | 5-8 |  |
| Лабораторийн тайлан бичилт, хамгаалалт | 30% | 9 |  |
| Нийт үнэлгээ /хувиар/ | | |  |
| Нийт үнэлгээ /оноогоор/ | | |  |

|  |  |  |
| --- | --- | --- |
| **ГАРЧИГ** | | |
|  | | |
| Лабораторийн хичээлийн хөтөлбөр | |  |
| Лабораторт тавигдах шаардлага, мөрдөх журам ба аюулгүй ажиллагааны  заавар | |  |
| Лабораторийн ажлууд | |  |
| Ажлын  дугаар | Ажлын нэр |  |
| 1 | Лабораторийн ажил №6 |  |
| 2 | Лабораторийн ажил №7 |  |

## Зургаадугаар лабораторийн ажил Тэмдэгт мөр ба массив

### Лабораторийн ажлын зорилго:

Уг лабораторын ажлаар тэмдэгт мөр ба бүхэл тоон массив ассемблер хэлэнд яаж дүрслэгддэг талаар сурна.

### Лабораторийн ажлын суралцахуйн үр дүнгүүд:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **д/д** | **Суралцахуйн үр дүнгүүд** | **Суралцахуйн үр дүнг илэрхийлэх**  **үйл үг** | **Суралцахуйн үр дүнгийн түвшин**  **(Блумын)** | **CLOs**  **хамаарал** |
| 1 | Тэг төгсгөлтэй тэмдэгт мөр |  |  |  |
| 2 | Бүхэл тоон массив |  |  |  |

**Ашиглах програм хангамж/техник хангамж, бусад хэрэглэгдэхүүнүүд:**

QTSpim программ хангамж болон Notepad хэрэглэгдэнэ.

### Онолын ойлголтууд:

#### Тэг төгсгөлтэй тэмдэгт мөр

Тэг төгсгөлтэй тэмдэгт мөр гэдэг нь төгсгөлийн байт нь тэг байт байх ASCII тэмдэгтүүдийн дараалал юм. Хоосон зайнууд нь ASCII space (0x20) -аар дүүргэгдэнэ. Харин хамгийн сүүлчийн байтаас хойшхи нь (0x00) байтуудаар дүүргэгдэнэ.

#### Бүхэл тоон массив

Бүхэл тоонуудын массив нь санах ойд байрлах бүхэл тоонуудын дараалал юм. Ассемблер хэлэнд бүхэл тоонуудын массивыг .word директивийн араас бүхэл тоонуудыг хооронд нь таслалаар зааглан бичиж зарладаг.

### Ажил гүйцэтгэх дараалал:

Жишээ кодуудыг ажилуулж үзсэний дараа даалгаваруудыг гүйцэтгэнэ.

### Суралцахуйн үр дүнг үнэлэх даалгаварууд:

#### Дасгал 1

Санах ойд дараах тэмдэгт мөрийг зарла.

.data

string: .asciiz “MIPS IS RISC and PIPELINED PrOCesSoR” Тэмдэгт мөрийн бүх том үсгүүдийг жижиг үсэг болгон хувиргадаг програм бич.

**A screenshot of a computer screen

Description automatically generated**

#### Дасгал 2

Санах ойд дараах тэмдэгт мөрийг зарла.

Санах ойд дараах тэмдэгт мөрийг зарла.

.data

string: asciiz "pentium is example of CISC processor"Дээр зарласан тэмдэгт мөрийн уг бүрийн эхний үсгийг том үсэг болгодог програм бич.

Програм ажилласны дараа дараах байдлаар харагдана

string: asciiz "Pentium Is Example Of CISC Processor"

### Үр дүн:

A screenshot of a computer

Description automatically generated

#### Дасгал 3

Санах ойд дараах бүхэл тооны массив зарла.

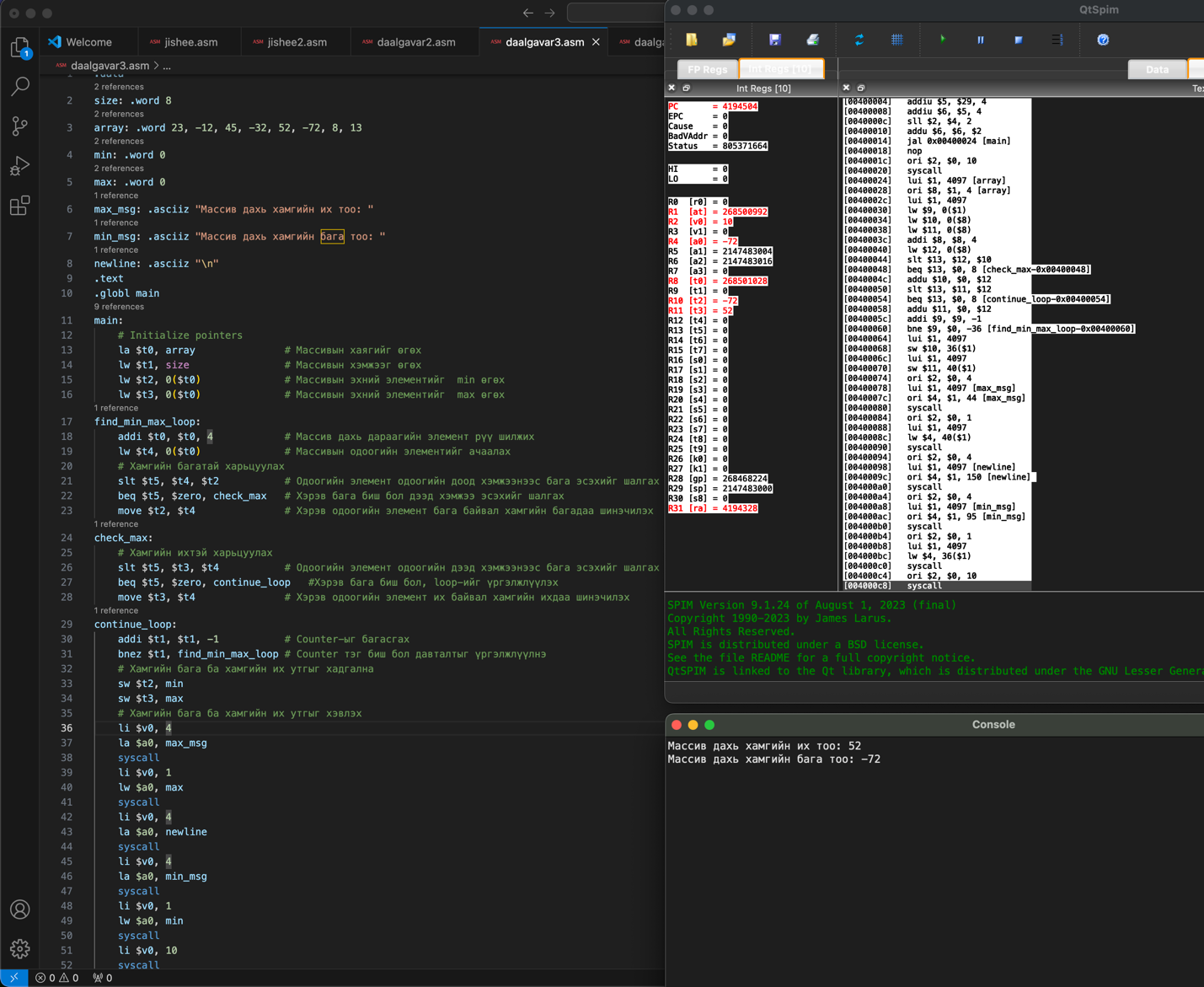
.data

size: .word 8

array: .word 23, -12, 45, -32, 52, -72, 8, 13

Массивын хамгийн бага ба хамгийн их элементийг олдог програм бич.

### Үр дүн:



#### Дасгал 4

Дараах бүхэл тооны массивыг зарла.

.data

size: .word 10

array: .word 2, 4, 7, 12, 34, 36, 42, 8, 57, 78

Хэрэв дарааллын зүүн талд орших элемент нь баруун талынхаа элементээс их байвал дарааллын хамгийн баруун талын элементээс эхлэн дарааллын элементүүдийн байрыг соль.

A screenshot of a computer program

Description automatically generated

A close-up of a computer screen

Description automatically generated

#### Дасгал 5

Санах ойд байрлаж байгаа 5 байт тооноос хичнээн ширхэг төгс тоо байгааг тоолж санах ойд хадгал. Төгс тоо гэдэг нь хуваагчуудын нийлбэр нь уг тоотой тэнцүү байдаг тоог хэлнэ. Жишээ нь 6 юм. 6=1+2+3. Мөн 28 төгс тоо юм.

### Ашиглах материал:

Лекц-6 материал

Лабораторын ажил 6 материал

**ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ**

**Мэдээлэл холбооны технологийн сургууль**

**A blue and yellow sign with a gear and drops

Description automatically generated**

**ЛАБОРАТОРИЙН АЖЛЫН**

**ТАЙЛАН**

**Доод түвшиний программчлал (F.NSA233)**

**2023-2024 оны хичээлийн жилийн Хавар**

**Лабораторийн ажлын нэр:** Лабораторийн Ажил №7

**Хичээл заасан багш:** Б.Мөнхбаяр

**Лабораторийн ажил гүйцэтгэсэн:** Оюутан: Б.Түшигт/В221870516/

Б.Мөнхбаяр/В221870521/

|  |  |  |  |
| --- | --- | --- | --- |
| **Авбал зохих нийт оноо: 5 оноо** | | | |
| **a** | | | |
| **Гүйцэтгэлийн шалгуур** | **Үнэлгээний**  **эзлэх хувь** | **CLOs**  **хамаарал** | **Багшийн**  **үнэлгээ** |
| Хичээлийн ирц/оролцоо | 10% | 4-9 |  |
| Доод түвшиний прораммчлалын симуляци болон  багцад дүн шинжилгээ хийх програмыг ашиглах | 10% | 4 |  |
| Лабораторийн заавар, дарааллын дагуу жишээг хийж  гүйцэтгэх | 10% | 5-8 |  |
| Лабораторийн заавар дээр өгөгдсөн даалгаврыг хий  гүйцэтгэх | 40% | 5-8 |  |
| Лабораторийн тайлан бичилт, хамгаалалт | 30% | 9 |  |
| Нийт үнэлгээ /хувиар/ | | |  |
| Нийт үнэлгээ /оноогоор/ | | |  |

## Долоодугаар лабораторийн ажил Extended assembler

### Лабораторийн ажлын зорилго:

Уг лабораторын ажлаар өргөтгөсөн ассемблер хэлний хийсвэр командуудтай танилцана.

### Лабораторийн ажлын суралцахуйн үр дүнгүүд:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **д/д** | **Суралцахуйн үр дүнгүүд** | **Суралцахуйн үр дүнг илэрхийлэх**  **үйл үг** | **Суралцахуйн үр дүнгийн түвшин**  **(Блумын)** | **CLOs**  **хамаарал** |
| 1 | Регистрийн мнемоник (mnemonic) нэрүүд |  |  |  |
| 2 | move псевдо команд |  |  |  |
| 3 | li псевдо команд |  |  |  |
| 4 | la псевдо команд |  |  |  |
| 5 | lw псевдо команд |  |  |  |
| 6 | sw псевдо команд |  |  |  |
| 7 | nop псевдо команд |  |  |  |

**Ашиглах програм хангамж/техник хангамж, бусад хэрэглэгдэхүүнүүд:**

QTSpim программ хангамж болон Notepad хэрэглэгдэнэ.

### Онолын ойлголтууд:

#### Өргөтгөсөн ассемблер

Тухайн процессорын үндсэн үйлдлүүд, операэндууд, болон хаягуудад харгалзах битийн төлвүүдийг симболуудаар илэрхийлэх боломжийг ассемблер хэл олгодог. Ассемблер хэлний команд шууд харгалзаж байдаг.

Өргөтгөсөн ассемблер хэл гэдэг нь MIPS -ийн бодит техник хангамж дахь командын бүлээс илүү баялаг командын бүл бүхий хийсвэр машины хэл юм. Өргөтгөсөн ассемблер хэлний ихэнх илэрхийллүүдийн цаана хэд хэдэн үндсэн машины командууд харгалзана.

Өргөтгөсөн ассемблерийн кодууд үндсэн ассемблерийн кодууд руу өргөтгөсөн ассемблерийн хөрвүүлэгчээр хөрвүүлэгдэнэ. Дараа нь үндсэн ассемблерийн кодууд машины кодууд руу үндсэн ассемблерийн хөрвүүлэгчээр хөрвүүлэгдсэнээр процессорын ойлгох машин код үүснэ.

#### Регистр мнемоник (mnemonic)

Ерөнхий зориулалтын регистерүүдийг програмд тусгай зориулалтаар ашиглах нь программ бичих асуудлыг хөнгөлдөг. Энэ нь програм бичигч будлихгүй байхын тулд хэрэглэнэ. Өргөтгөсөн ассемблер хэлэнд регистрүүдийн мнемоник нэрүүдийг хэрэглэх ба ингэснээр тухайн регистр ямар зорилгоор хэрэглэгдэж байгааг тодорхой болгодог.

#### Псевдо командууд (Pseudoinstructions)

Псевдо буюу хийсвэр команд гэдэг нь үндсэн ассемблер хэлний нэг болон түүнээс дээш командуудаар биелэгдэх өргөтгөсөн ассемблер хэлний нэг команд юм.

#### Санах ойн хаяг ба хаяган дахь өгөгдөл

Нэг үг (word) санах ойн хэмжээ 32 бит байдаг. Мөн нэг үг санах ойг 32 битээр хаягладаг. Эдгээр хоёр өгөгдөл нь (санах ойн хаяг ба санах ойн хэмжээ) хоёул 32 битийн регистрт ачаалагдаж болно.

ҮГ санах ойд хадгалагдаж байгаа өгөгдлийг регистрт ачаалахдаа lw командыг ашигладаг.

Харин ҮГ санах ойн хаягийг регистрт ачаалах тохиолдолд la командыг ашиглана.

#### ҮГ-ийг хадгалах команд

sw мнемоник үндсэн болон өргөтгөсөн ассемблер хэлний аль алинд хэрэглэгдэнэ. sw псевдо команд нь регистрийн утгыг санах ойд хадгалахдаа нэг командаар хадгалагна.

### Ажил гүйцэтгэх дараалал:

Жишээ кодуудыг ажилуулж үзсэний дараа даалгаваруудыг гүйцэтгэнэ.

### Суралцахуйн үр дүнг үнэлэх даалгаварууд:

#### Даалгавар 1

Дараах илэрхийллийг бод ax^3+ bx^2 +cx + d

.data

x: .word 7

a: .word -3

b: .word 3

c: . word 9 d: .word -24

result: .word 0

Коэффициентүүдийг санах ойгоос регистрт ачаалах, үр дүнг санах ойд хадгалахдаа lw, sw псевдо командуудыг хэрэглэнэ. Харгалзуулан нэмээд үр дүнг result массивт хадгалах давталт бичнэ.

### Үр дүн:

A screenshot of a computer program

Description automatically generatedA screenshot of a computer code

Description automatically generated

#### Даалгавар 2

Массивын элементүүдийг харгалзуулан нэмэх Адил хэмжээтэй гурван массив зарла.

.data

size .word 7

array1: .word -30, -23, 56, -43, 72, -18, 71

array2: .word 45, 23, 21, -23, -82, 0, 69

result: .word 0, 0, 0, 0, 0, 0, 0

Үндсэн регистрт массив тус бүрийн эхлэх хаягийг la командаар ачаална. Эхний хоёр массивын элемент бүрийг харгалзуулан нэмээд үр дүнг result массивт хадгалах давталт бичнэ.

### Asm код:



### Үр дүн:

**A white rectangular object with a black border

Description automatically generated**

#### Даалгавар 3

Тэмдэгт мөр харьцуулах Тэмдэгт мөр зарлана.

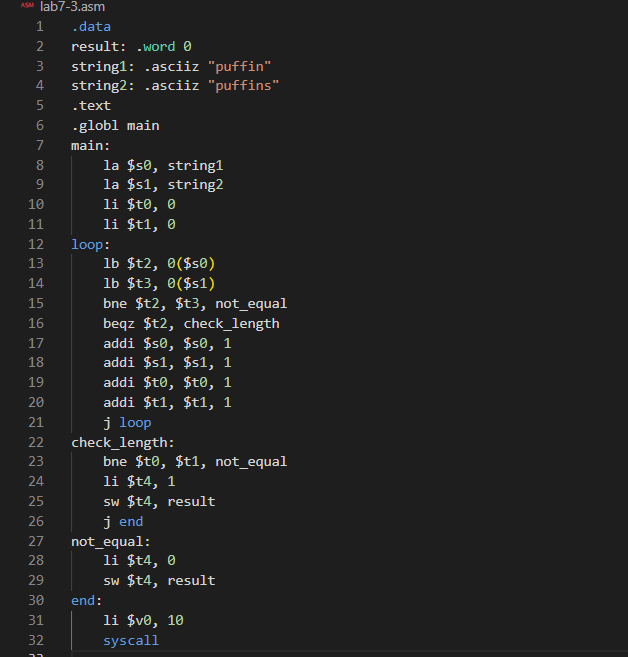
.data result: .word 0

string1: .asciiz “puffin” string2: .asciiz “puffins”

Тэмдэгт мөр тус бүрийн үндсэн хаягийг la командаар регистрт ачаална. Хэрэв тэмдэгт мөрүүд адилхан урттай байвал result -д 1-ийг эсрэг тохиолдолд 0-ийг бичдэг програм бич.

### Asm код:

.data

A screenshot of a computer

Description automatically generated

### Үр дүн:

**Ашиглах материал:**

Лекц-7 материал

Лабораторын ажил 7 материал

**ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ**

**Мэдээлэл холбооны технологийн сургууль**

**A blue and yellow sign with a gear and drops

Description automatically generated**

**ЛАБОРАТОРИЙН АЖЛЫН**

**ТАЙЛАН**

**Доод түвшиний программчлал (F.NSA233)**

**2023-2024 оны хичээлийн жилийн Хавар**

**Лабораторийн ажлын нэр:** Лабораторийн Ажил №8

**Хичээл заасан багш:** Б.Мөнхбаяр

**Лабораторийн ажил гүйцэтгэсэн:** Оюутан: Б.Түшигт/В221870516/

Б.Мөнхбаяр/В221870521/

|  |  |  |  |
| --- | --- | --- | --- |
| **Авбал зохих нийт оноо:** | **5 оноо** | | |
|  | | | |
| **Гүйцэтгэлийн шалгуур** | **Үнэлгээний**  **эзлэх хувь** | **CLOs**  **хамаарал** | **Багшийн**  **үнэлгээ** |
| Хичээлийн ирц/оролцоо | 10% | 4-9 |  |
| Доод түвшиний прораммчлалын симуляци болон  багцад дүн шинжилгээ хийх програмыг ашиглах | 10% | 4 |  |
| Лабораторийн заавар, дарааллын дагуу жишээг хийж  гүйцэтгэх | 10% | 5-8 |  |
| Лабораторийн заавар дээр өгөгдсөн даалгаврыг хий  гүйцэтгэх | 40% | 5-8 |  |
| Лабораторийн тайлан бичилт, хамгаалалт | 30% | 9 |  |
| Нийт үнэлгээ /хувиар/ | | |  |
| Нийт үнэлгээ /оноогоор/ | | |  |

## Наймдугаар лабораторийн ажил

**Оролт ба гаралт**

### Лабораторийн ажлын зорилго:

Уг лабораторийн ажлаар SPIM -ийн trap handler -т агуулагдах сервисүүдийг ашиглахад суралцана.

### Лабораторийн ажлын суралцахуйн үр дүнгүүд:

Энэ лабораторийн ажлыг гүйцэтгэснээр оюутан дараах чадваруудтай болсон байна.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **д/д** | **Суралцахуйн үр дүнгүүд** | **Суралцахуйн үр дүнг илэрхийлэх**  **үйл үг** | **Суралцахуйн үр дүнгийн түвшин**  **(Блумын)** | **CLOs**  **хамаарал** |
| 1 | (программыг дуусгах) halt program |  |  |  |
| 2 | (тэмдэгт мөрийг хэвлэх) print string |  |  |  |
| 3 | (тэмдэгт мөрийг унших) read string |  |  |  |
| 4 | (бүхэл тоог хэвлэх) print integer |  |  |  |
| 5 | (бүхэл тоог унших) read integer |  |  |  |

### Ашиглах програм хангамж/техник хангамж, бусад хэрэглэгдэхүүнүүд:

Notepad болон QTSpim программ хангамжууд ашиглагдана.

### Онолын ойлголтууд:

#### SPIM Exception Handler

Компьютерийн систем нь үйлдлийн системийн удирдлага дор ажиллаж байдаг ба хэрэглээний программууд оролт, гаралт болон бусад системийн даалгаварыг үйлдлийн системийн сервисүүдийг ашиглан гүйцэтгэдэг.

SPIM симулятор үйлдлийн систем агуулаагүй ч ассемблер программ бичихэд тус дөхөм болохоор сервисүүдийг багтаасан жижиг trap handler -ийг ажиллуулах боломжтойгоор зохиогдсон.

#### syscall

Ассемблер хэлний програмууд үйлдлийн системийн сервисүүдийг syscall командын тусламжтайгаар хүсдэг.

syscall # үйлдлийн системээс тухайн сервисийг гүйцэтгэж өгөхийг хүснэ.

Үйлдлийн систем бүр уг командыг ялгаатай аргаар хэрэглэдэг ч SPIM trap handler дараах байдлаар хэрэглэнэ

li $v0,code # $v0 регистрт ҮС-ийн тухайн сервисийн "код"-ын дугаарыг ачаална. # сервисийн оролтын параметрүүдийг ачаална....

# $a0, $a1 эсвэл $f12 регистрүүд # Үйлдлийн системээс хүснэ syscall

# буцаж ирэх утгууд $v0 болон $f0

Ялгаатай сервисүүд ялгаатай регистрүүдийг ашиглах ба сервис болгон утга буцаах албагүй.

**Тэмдэгт мөр хэвлэх** SPIM-ийн тэмдэгт мөр хэвлэх trap handler нь тэг төгсгөлтэй тэмдэгт мөрийг дэлгэц (console) -нд хэвлэнэ.

Тэмдэгт мөрийн хаягийг $a0 регистрт ачаална. li $v0,4

#code 4 print string la $a0,string

# $a0 address of the string

syscall # үйлдлийн системээс сервисийг хүснэ

.data

string: .asciiz "Hello SPIM!\n"

Уг сервис тэмдэгт мөрийг төгсгөлийн тэг байт гарч иртэл консолд хэвлэнэ. Хэрэв шинэ мөрнөөс хэвлэхийг хүсвэл тэмдэгт мөрийн төгсгөлд "п'-ийг оруулна

#### Тэмдэгт мөрийг унших

Тэмдэгт мөр унших сервис нь гарнаас дарагдах тэмдэгт мөр бүрийг уншина. li $v0,8

# code 8 == тэмдэгт мөр унших la $a0, buffer

# $a0 buffer санах ойн хаяг li $a1,16

# $a1 = buffer санах ойн урт syscall

.data

buffer: .space 16

# 16 байтыг нөөцлөнө

Программыг дуусгахдаа exit cсервисийн код болох 10-ийг $v0 регистрт ачаалаад syscall командаар системийн сервисийг хүснэ.

exit

# code 10 li $v0,10 syscall

#### Бүхэл тоог унших, хэвлэх

Бүхэл тоог унших сервис нь гарнаас дарагдах '0', '1', .., '9' тоонуудын ASCII кодыг '-' болон '+' тэмдэгтийн хамт уншина. $v0 регистрт бүхэл тооны 32 битийн хоёртын гүйцээлтийн дүрслэлийг буцаана.

li $v0,5 # code 5 read integer Syscall

Бүхэл тоог хэвлэх сервис нь $a0 регистр дэх бүхэл тоог Дэлгэцэнд хэвлэнэ. li $v0,1 # code 1 бүхэл тоог хэвлэх

li $a0,int # $a0 бүхэл тоо

# Convert the 32-bit integer into characters.

syscall

# Print the character to the monitor.

### Суралцахуйн үр дүнг үнэлэх даалгаварууд:

Фарангет ба Цельсийн хөрвүүлэгч. Хэрэглэгч F эсвэл С оруулна. ("Fahrenheit" эсвэл "Celsius"). Хэрэглэгчийн оруулсан фарангет эсвэл цельсийн нэгж бүхий температурын утгыг цельс эсвэл фарангет нэгжид хөрвүүлнэ. F= (9/5)C+32, C = (5/9)(F-32) Хэрэглэгч гарнаас "Q"-г дарахад программ дуусна.

Enter Scale : F

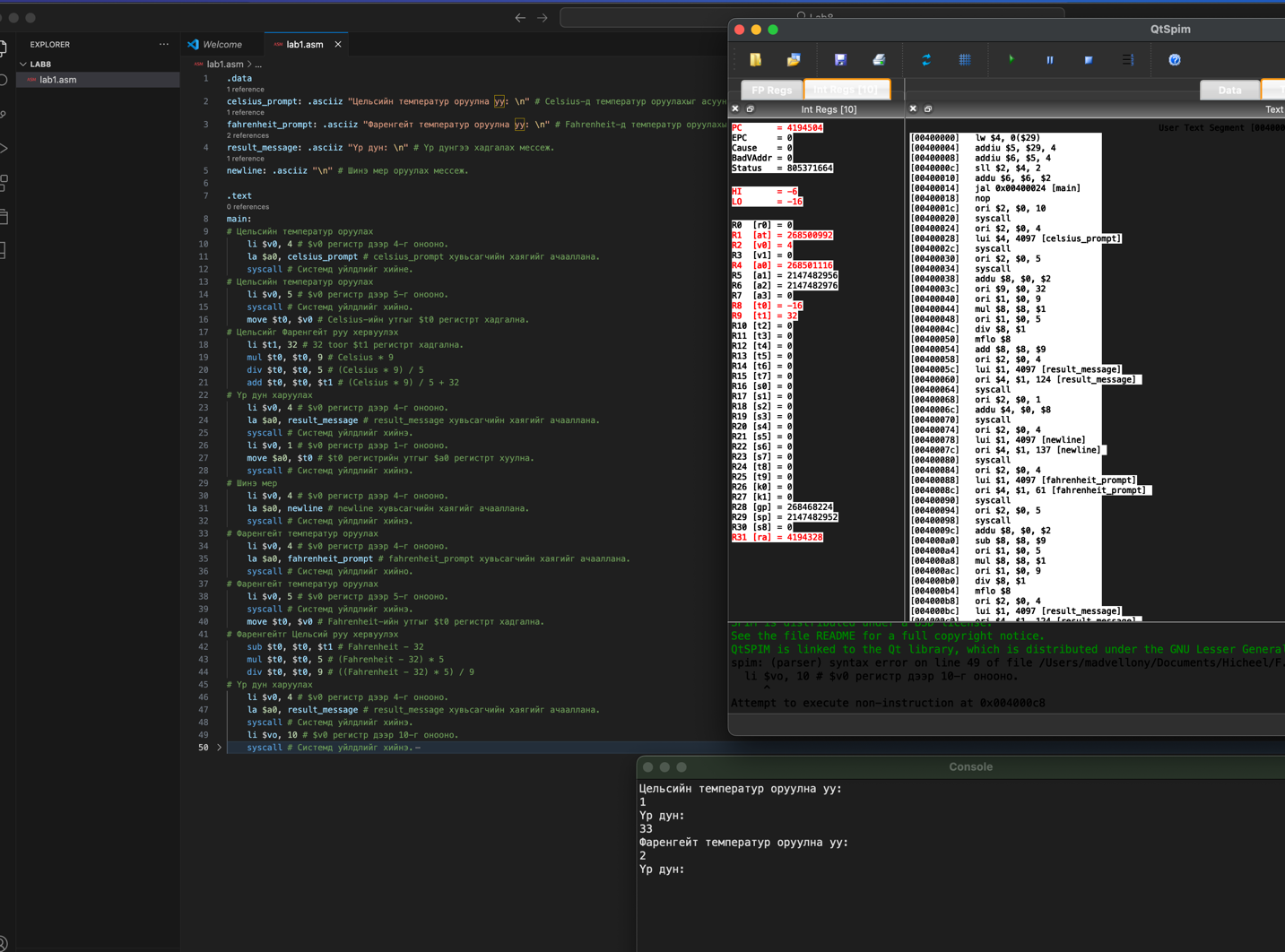
Enter Temperature: 32 Celsius Temperature: OC Enter Scale : C

Enter Temperature: 100 Fahrenheit Temperature: 212F Enter Scale : Q

Done

Гараас оруулсан 5 тооноос анхны тоо хэд байгааг тоолж, үр дүнг дэлгэцэнд хэвлэ.

**Asm код:**



**Үр дүн:**

A screenshot of a computer

Description automatically generated

**Ашиглсан материал:**

Лекц-8 материал

Лабораторийн ажил 8 материал

**ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ**

**Мэдээлэл холбооны технологийн сургууль**

**A blue and yellow sign with a gear and drops

Description automatically generated**

**ЛАБОРАТОРИЙН АЖЛЫН**

**ТАЙЛАН**

**Доод түвшиний программчлал (F.NSA233)**

**2023-2024 оны хичээлийн жилийн Хавар**

**Лабораторийн ажлын нэр:** Лабораторийн Ажил №9

**Хичээл заасан багш:** Б.Мөнхбаяр

**Лабораторийн ажил гүйцэтгэсэн:** Оюутан: Б.Түшигт/В221870516/

Б.Мөнхбаяр/В221870521/

|  |  |  |  |
| --- | --- | --- | --- |
| **Авбал зохих нийт оноо:** | **5 оноо** | | |
|  | | | |
| **Гүйцэтгэлийн шалгуур** | **Үнэлгээний**  **эзлэх хувь** | **CLOs**  **хамаарал** | **Багшийн**  **үнэлгээ** |
| Хичээлийн ирц/оролцоо | 10% | 4-9 |  |
| Доод түвшиний прораммчлалын симуляци болон  багцад дүн шинжилгээ хийх програмыг ашиглах | 10% | 4 |  |
| Лабораторийн заавар, дарааллын дагуу жишээг хийж  гүйцэтгэх | 10% | 5-8 |  |
| Лабораторийн заавар дээр өгөгдсөн даалгаврыг хий  гүйцэтгэх | 40% | 5-8 |  |
| Лабораторийн тайлан бичилт, хамгаалалт | 30% | 9 |  |
| Нийт үнэлгээ /хувиар/ | | |  |
| Нийт үнэлгээ /оноогоор/ | | |  |

## Есдүгээр лабораторийн ажил

**Subroutine**

### Лабораторийн ажлын зорилго:

Бид уг лабораторийн ажлаар Subroutine буюу дэд функцийн талаар үзнэ.

### Лабораторийн ажлын суралцахуйн үр дүнгүүд:

Энэ лабораторийн ажлыг гүйцэтгэснээр оюутан дараах чадваруудтай болсон байна.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **д/д** | **Суралцахуйн үр дүнгүүд** | **Суралцахуйн үр дүнг илэрхийлэх**  **үйл үг** | **Суралцахуйн үр дүнгийн түвшин**  **(Блумын)** | **CLOs**  **хамаарал** |
| 1 | Subroutine дуудалт |  |  |  |
| 2 | Caller ба Callee routine |  |  |  |
| 3 | Jal ба jr команд |  |  |  |
| 4 | $га регистр |  |  |  |
| 5 | Subroutine-д ашиглагддаг регистр |  |  |  |

### Ашиглах програм хангамж/техник хангамж, бусад хэрэглэгдэхүүнүүд:

Notepad болон QTSpim программ хангамжууд ашиглагдана.

### Онолын ойлголтууд:

#### Энгийн Subroutine Холбоос

Бүх дээд түвшний хэл нь subroutine-тэй (заримдаа процедур, функц, эсвэл метод гэж нэрлэдэг). Subroutine нь өөртөө үйлдэл агуулж болох кодын логик хуваалт юм. Жишээ нь sine функц нь ихэвчлэн subroutine-ээр хэрэгждэг. Энэ нь програмд хэрэг болж ашиглагдах үйлдэл болж тооцогддог.

#### Callers ба Callees

Үндсэн процедур эхлэх хэрэгтэй үед ("дуудах") дэд процедур нь ј командтай хэрэгжинэ. Дэд процедурын төгсгөлд удирдлага өөр Ј командтай буцаж ирнэ. Дэд процедур нь зөвхөн нэг цэг дээр дуудагдах ба үүний дараа хэд хэдэн командын хаяг руу буцна. Дэд процедур нь зөвхөн үндсэн програм дээр хэрэглэгдэх ба учир нь хуучин байрлал дээр буцаж ирдэг. Дэд процедурын дуудалт нь үндсэн процедур удирдлагаа дэд процедур руу шилжүүлэх үед хэрэгжинэ. Энэ үндсэн процедурыг СALLER буюу ДУУДАГЧ харин дэд процедурыг CALLEE буюу ДУУДАГДАГЧ ГЭЖ нэрлэж болох юм. Дэд процедур нь удирдлагаа үндсэн процедур руу (CALLER) шилжүүлэх үед буцах үйлдэл хэрэгждэг. CALLEE нь ажиллагаагаа дуусгахад өөрийн

CALLER pyyrаа удирлагаа буцааж өгнө.

#### Олон дуудалт, нэг буцалт

Код дээр хэд хэдэн газар олон ашиглагдах дэд процедурыг дуудах замаар үндсэн процедурыг бичнэ. Гэвч дэд процедур дуудагдах бүрт зөвхөн нэг газар руу удирдлага буцна.

Өмнө нь техник хангамжийн хувьд дэд процедурын тухай ойлголт байсангүй. Үүнийг хэрэгжүүлэх арга зам нь мөн тодорхой биш байсан.

Дэд процедур дуудагдаж дуусахад түүн рүү буцах хаяг илгээх замаар асуудлыг шийджээ. Дэд процедур төгсөхөд удирдлагаа тухайн буцах хаяг руу шилжүүлдэг байна.

"буцах хаяг руу удирдлагаа буцаана" гэдэг нь дэд процедур нь ПТ (Програм Тоолуур)-ыг буцах хаягтай цуг ачаална гэсэн үг юм. Машин циклийн дараагийн команд нь тухайн хаягаас эхэлж ажиллана

#### Jal команд

Холбоосонд (linkage) ашиглагддаг регистр нь $31 ба энэ нь өргөтгөсөн ассемблерт Sra гэж нэрлэгдэнэ. Энэ нь дэд процедурын буцах хаягыг хадгалж байдаг. Ѕга регистр рүү буцах хаягыг хийдэг команд нь jal юм.

Регистр $31 нь хоёр “ерөнхий зориулалтын регистрүүдийн" нэг нь юм (нөгөө нь $0.) jal команд ба регистр $31 нь дэд процедуруудыг нэмэгдүүлэхэд техник хангамжийн дэмжлэгээр хангагдсан байдаг.

jal хэрхэн ажилладагийг ойлгохын тулд машин циклийг ажиглая. MIPS-н төгсгөлгүй цикл 3 үндсэн алхамаар дамждаг. Цикл бүр нэг машин команд гүйцэтгэнэ.

jal команд нь машин циклийн ажиллагааны үед дараах үйлдлийг хийнэ: jal sub #$ra-PC+8 Sra <- address 8 bytes away from the jal

#PC-sub load the PC with the subroutine entry point

Машин циклийн дунд шатанд ПТ-г 4-өөр нэмэгдүүлнэ. Энэ үед ПТ ја командын дараах командын хаягыг агуулна. Одоо jal командын тухайн хаяг руу 4-г нэмэж үр дүнг Ѕга-д хийх ажиллагаа явагдана. Ингээд Ѕга нь jal командын дараах хоёр дах командын хаягыг агуулна.

Буцах зөв хаяг нь "jal хаяг дээр нэмэх нь найм" юм. Учир нь: (і) Дэд процедураас jal команд руу буцахад осолтой (дахин ажиллаж болзошгүй), мөн (ii) jal-ийн дараах команд нь branch delay slot байна.

#### Jr команд

jr команд нь удирдлагыг дуудагч руу буцаана. Энэ команд $га дах утгыг ПТ руу хуулна: jr $ra

# PC <- $ra Usually you

think of this as "jumping to the address in $ra."

Командыг илүү ерөнхий болгохын тулд $га-аас бусад аль ч регистртэй хамт ашиглаж болно. Бүх үсрэлтийн болон салаалах командуудын адил, јг команд нь branch delay-г дагана.

Диаграм дээр jal командаар $га-д ачаалагдсан буцах хаягаар буцаж байгаа дэд процедурыг харуулав.

**Регистрийн хэрэглээ** Програм хангамжийн тохиргоогоор регистрүүд нь тус тусдаа үүрэгтэй байдаг:

$t0 St9 Дэд процедураас эдгээрийг өөрчлөхөд чөлөөтэй.

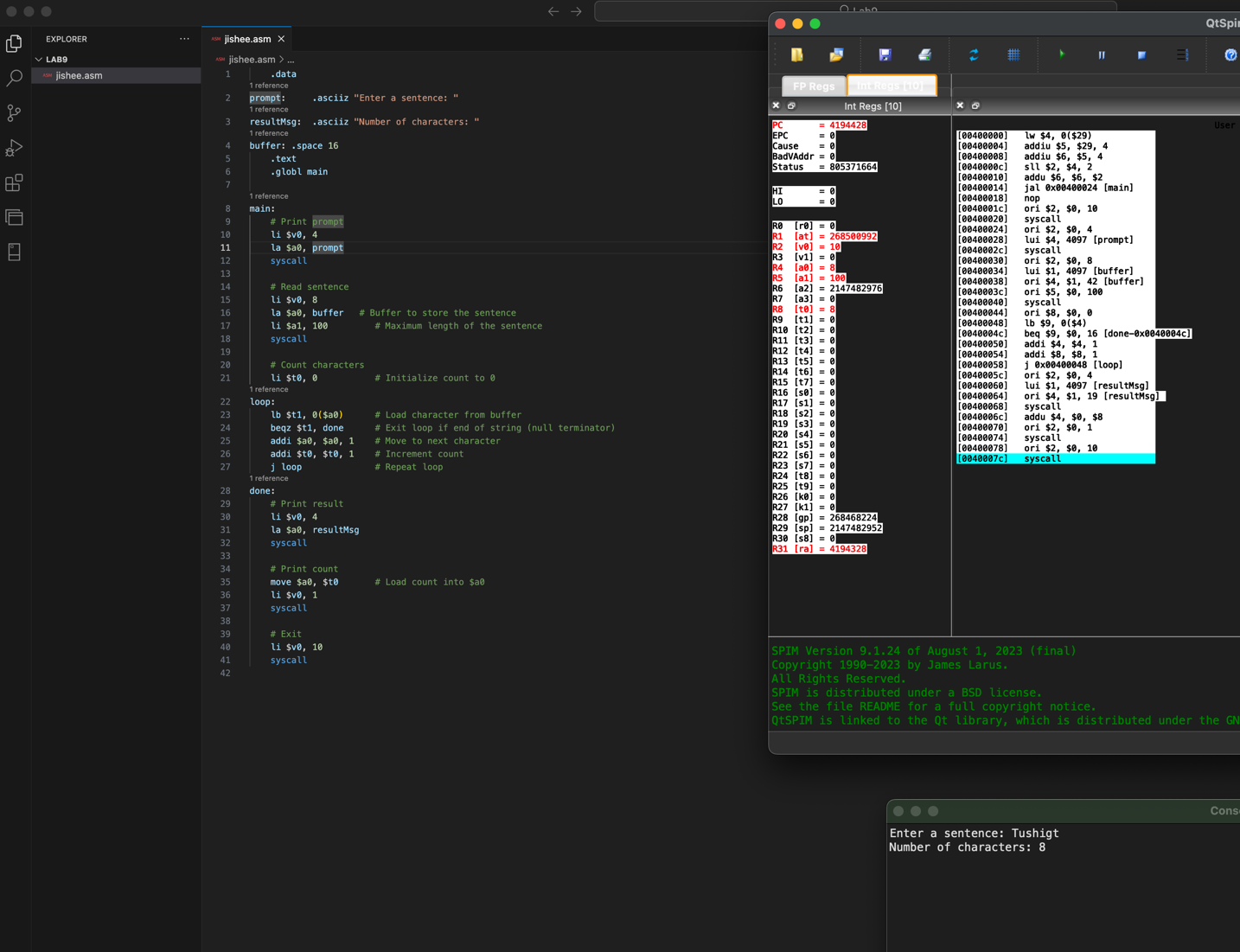
$50 - $s7 Дэд процедур эдгээрийг өөрчлөх ёсгүй. $a0 $a3 Эдгээр регистрүүд дэд процедурын аргументуудыг агуулж байна. Дэд процедур эдгээрийг өөрчлөх боломжтой.

$v0 - $v1 Эдгээр регистрүүд дэд процедураас буцаж ирсэн утгыг хадгална.

### Суралцахуйн үр дүнг үнэлэх даалгаварууд:

Гарнаас тэмдэгт мөр авдаг дэд процедуурыг бич. Хэрэглэгч нийт 4 удаа тэмдэгт мөрийг оруулах ба үндсэн програм нь нийт хичнээн тэмдэгт байгааг тоолж дэлгэцэнд тэмдэгтийн тоог хэвлэ.

**Asm код:**



**Үр дүн:**

A screenshot of a computer

Description automatically generated

**Ашигласан материал:**

Лекц-9 материал

Лабораторын ажил 9 материал