



**Монгол улсын их сургууль**  
**Мэдээллийн технологи, электроникийн сургууль**  
**Мэдээлэл , компьютерийн ухааны тэнхим**

## **Семинарын даалгавар №5**

Сэдэв: Протокол

Гүйцэтгэсэн: Програмын хангамж-3 Э.Түвшин-Эрдэнэ (23b1num0869)

Шалгасан багш:.....Д.Тэмүүжин

Улаанбаатар хот

2025 он

## Дасгал 1

| Хаяг  | Утга | Регистр | Утга  |
|-------|------|---------|-------|
| 0x100 | 0xFF | %rax    | 0x100 |
| 0x104 | 0xAB | %rcx    | 0x1   |
| 0x108 | 0x13 | %rdx    | 0x3   |
| 0x10C | 0x11 |         |       |

| Операнд       | Утга  |
|---------------|-------|
| %rax          | 0x100 |
| 0x104         | 0xAB  |
| \$0x108       | 0x108 |
| (%rax)        | 0xFF  |
| 4(%rax)       | 0xAB  |
| 9(%rax, %rdx) | 0x11  |
| 0xFC(,%rcx,4) | 0xFF  |
| (%rax,%rdx,4) | 0x11  |

## Дасгал 2

|      |                    |
|------|--------------------|
| movl | %eax, (%rsp)       |
| movw | (%rax), %dx        |
| movb | \$0xFF, %b1        |
| movb | (%rsp,%rdx,4), %d1 |
| movq | (%rdx), %rax       |
| movw | %dx, (%rax)        |

## Дасгал 3

|                      |  |
|----------------------|--|
| Movb \$0xF, (%ebx)   | ebx ийг хаягийн регистрээр ашиглаж болохгүй                    |
| Movl %rax, (%rsp)    | 64 битийн утгыг 32 битийн санах ой руу зөөж байна              |
| Movw (%rax), 4(%rsp) | Source болон destination хоёулаа санах ойн хаяг байж болохгүй. |
| Movb %al, %sl        | sl гэдэг регистр байдаггүй                                     |
| Movq %rax, \$0x123   | Destination буруу  |
| Movl %eax, %rdx      | 32 битийн утгыг 64 битийн санах ой руу зөөж байна              |
| Movl %si, 8(%rbp)    | Заавар болон source-ийн бит зөрж байна.                        |

## Дасгал 4

Disassembly of section .init:

0000000000001000 <\_init>:

```

1000:  f3 0f 1e fa      endbr64
1004:  48 83 ec 08      sub   $0x8,%rsp
1008:  48 8b 05 d9 2f 00 00  mov   0x2fd9(%rip),%rax    # 3fe8
<__gmon_start__@Base>
100f:  48 85 c0         test  %rax,%rax
1012:  74 02           je    1016 <_init+0x16>
1014:  ff d0          call  *%rax
1016:  48 83 c4 08      add   $0x8,%rsp
101a:  c3             ret

```

Disassembly of section .plt:

0000000000001020 <.plt>:

```

1020:  ff 35 9a 2f 00 00  push  0x2f9a(%rip)    # 3fc0
<_GLOBAL_OFFSET_TABLE_+0x8>
1026:  ff 25 9c 2f 00 00  jmp   *0x2f9c(%rip)    # 3fc8
<_GLOBAL_OFFSET_TABLE_+0x10>
102c:  0f 1f 40 00      nopl  0x0(%rax)
1030:  f3 0f 1e fa      endbr64
1034:  68 00 00 00 00  push  $0x0
1039:  e9 e2 ff ff ff  jmp   1020 <_init+0x20>
103e:  66 90          xchg  %ax,%ax

```

Disassembly of section .plt.got:

0000000000001040 <\_\_cxa\_finalize@plt>:

```

1040:  f3 0f 1e fa      endbr64

```

## Компьютерийн системийн зарчим 1

```
1044: ff 25 ae 2f 00 00    jmp  *0x2fae(%rip)    # 3ff8
<__cxa_finalize@GLIBC_2.2.5>
```

```
104a: 66 0f 1f 44 00 00    nopw 0x0(%rax,%rax,1)
```

Disassembly of section .plt.sec:

0000000000001050 <printf@plt>:

```
1050: f3 0f 1e fa          endbr64
```

```
1054: ff 25 76 2f 00 00    jmp  *0x2f76(%rip)    # 3fd0 <printf@GLIBC_2.2.5>
```

```
105a: 66 0f 1f 44 00 00    nopw 0x0(%rax,%rax,1)
```

Disassembly of section .text:

0000000000001060 <\_start>:

```
1060: f3 0f 1e fa          endbr64
```

```
1064: 31 ed               xor  %ebp,%ebp
```

```
1066: 49 89 d1            mov  %rdx,%r9
```

```
1069: 5e                 pop  %rsi
```

```
106a: 48 89 e2            mov  %rsp,%rdx
```

```
106d: 48 83 e4 f0         and  $0xfffffffffff0,%rsp
```

```
1071: 50                 push %rax
```

```
1072: 54                 push %rsp
```

```
1073: 45 31 c0            xor  %r8d,%r8d
```

```
1076: 31 c9              xor  %ecx,%ecx
```

```
1078: 48 8d 3d ff 00 00 00 lea  0xff(%rip),%rdi    # 117e <main>
```

```
107f: ff 15 53 2f 00 00    call *0x2f53(%rip)    # 3fd8
<__libc_start_main@GLIBC_2.34>
```

```
1085: f4                 hlt
```

```
1086: 66 2e 0f 1f 84 00 00 cs nopw 0x0(%rax,%rax,1)
```

```
108d: 00 00 00
```

## Компьютерийн системийн зарчим 1

0000000000001090 <deregister\_tm\_clones>:

```
1090: 48 8d 3d 79 2f 00 00 lea 0x2f79(%rip),%rdi    # 4010 <__TMC_END__>
1097: 48 8d 05 72 2f 00 00 lea 0x2f72(%rip),%rax    # 4010 <__TMC_END__>
109e: 48 39 f8             cmp  %rdi,%rax
10a1: 74 15               je   10b8 <deregister_tm_clones+0x28>
10a3: 48 8b 05 36 2f 00 00 mov 0x2f36(%rip),%rax    # 3fe0
<_ITM_deregisterTMCloneTable@Base>
10aa: 48 85 c0             test %rax,%rax
10ad: 74 09               je   10b8 <deregister_tm_clones+0x28>
10af: ff e0               jmp  *%rax
10b1: 0f 1f 80 00 00 00 00 nopl 0x0(%rax)
10b8: c3                 ret
10b9: 0f 1f 80 00 00 00 00 nopl 0x0(%rax)
```

00000000000010c0 <register\_tm\_clones>:

```
10c0: 48 8d 3d 49 2f 00 00 lea 0x2f49(%rip),%rdi    # 4010 <__TMC_END__>
10c7: 48 8d 35 42 2f 00 00 lea 0x2f42(%rip),%rsi    # 4010 <__TMC_END__>
10ce: 48 29 fe             sub  %rdi,%rsi
10d1: 48 89 f0             mov  %rsi,%rax
10d4: 48 c1 ee 3f          shr  $0x3f,%rsi
10d8: 48 c1 f8 03          sar  $0x3,%rax
10dc: 48 01 c6             add  %rax,%rsi
10df: 48 d1 fe             sar  $1,%rsi
10e2: 74 14               je   10f8 <register_tm_clones+0x38>
```

## Компьютерийн системийн зарчим 1

```
10e4: 48 8b 05 05 2f 00 00  mov  0x2f05(%rip),%rax    # 3ff0
<__ITM_registerTMCloneTable@Base>

10eb: 48 85 c0              test  %rax,%rax

10ee: 74 08                je   10f8 <register_tm_clones+0x38>

10f0: ff e0                jmp  *%rax

10f2: 66 0f 1f 44 00 00    nopw  0x0(%rax,%rax,1)

10f8: c3                  ret

10f9: 0f 1f 80 00 00 00 00  nopl  0x0(%rax)

0000000000001100 <__do_global_dtors_aux>:

1100: f3 0f 1e fa          endbr64

1104: 80 3d 05 2f 00 00 00  cmpb  $0x0,0x2f05(%rip)    # 4010 <__TMC_END__>

110b: 75 2b                jne  1138 <__do_global_dtors_aux+0x38>

110d: 55                  push  %rbp

110e: 48 83 3d e2 2e 00 00  cmpq  $0x0,0x2ee2(%rip)    # 3ff8
<__cxa_finalize@GLIBC_2.2.5>

1115: 00

1116: 48 89 e5              mov  %rsp,%rbp

1119: 74 0c                je   1127 <__do_global_dtors_aux+0x27>

111b: 48 8b 3d e6 2e 00 00  mov  0x2ee6(%rip),%rdi    # 4008 <__dso_handle>

1122: e8 19 ff ff ff       call 1040 <__cxa_finalize@plt>

1127: e8 64 ff ff ff       call 1090 <deregister_tm_clones>

112c: c6 05 dd 2e 00 00 01  movb  $0x1,0x2edd(%rip)    # 4010 <__TMC_END__>

1133: 5d                  pop  %rbp

1134: c3                  ret

1135: 0f 1f 00              nopl  (%rax)

1138: c3                  ret
```

## Компьютерийн системийн зарчим 1

1139: 0f 1f 80 00 00 00 00 nopl 0x0(%rax)

0000000000001140 <frame\_dummy>:

1140: f3 0f 1e fa endbr64

1144: e9 77 ff ff jmp 10c0 <register\_tm\_clones>

0000000000001149 <fact>:

1149: f3 0f 1e fa endbr64

114d: 55 push %rbp

114e: 48 89 e5 mov %rsp,%rbp

1151: 48 83 ec 20 sub \$0x20,%rsp

1155: 89 7d ec mov %edi,-0x14(%rbp)

1158: 83 7d ec 00 cmpl \$0x0,-0x14(%rbp)

115c: 75 07 jne 1165 <fact+0x1c>

115e: b8 01 00 00 00 mov \$0x1,%eax

1163: eb 17 jmp 117c <fact+0x33>

1165: 8b 45 ec mov -0x14(%rbp),%eax

1168: 83 e8 01 sub \$0x1,%eax

116b: 89 c7 mov %eax,%edi

116d: e8 d7 ff ff call 1149 <fact>

1172: 89 45 fc mov %eax,-0x4(%rbp)

1175: 8b 45 fc mov -0x4(%rbp),%eax

1178: 0f af 45 ec imul -0x14(%rbp),%eax

117c: c9 leave

117d: c3 ret

000000000000117e <main>:

117e: f3 0f 1e fa endbr64



## Компьютерийн системийн зарчим 1

```
1182:  55          push  %rbp
1183:  48 89 e5     mov   %rsp,%rbp
1186:  48 83 ec 10   sub   $0x10,%rsp
118a:  bf 05 00 00 00  mov   $0x5,%edi
118f:  e8 b5 ff ff ff call  1149 <fact>
1194:  89 45 fc     mov   %eax,-0x4(%rbp)
1197:  8b 45 fc     mov   -0x4(%rbp),%eax
119a:  89 c6        mov   %eax,%esi
119c:  48 8d 05 61 0e 00 00 lea   0xe61(%rip),%rax    # 2004 <_IO_stdin_used+0x4>
11a3:  48 89 c7     mov   %rax,%rdi
11a6:  b8 00 00 00 00  mov   $0x0,%eax
11ab:  e8 a0 fe ff ff call  1050 <printf@plt>
11b0:  b8 00 00 00 00  mov   $0x0,%eax
11b5:  c9          leave
11b6:  c3          ret
```

Disassembly of section .fini:

00000000000011b8 <\_fini>:

```
11b8:  f3 0f 1e fa   endbr64
11bc:  48 83 ec 08   sub   $0x8,%rsp
11c0:  48 83 c4 08   add   $0x8,%rsp
11c4:  c3          ret
```

## Дасгал 5

```
void decode1(long *xp, long *yp, long *zp)
{
    long val_x = *xp;
    long val_y = *yp;
    long val_z = *zp;

    *yp = val_x;
    *zp = val_y;
    *xp = val_z;
}
```