

Зертханалық жұмыс №3

Пән: Компьютер архитектурасы және операциялар келісімділігі

Тақырыбы: MIPS шартты/шартсыз операторлармен бағдарлама жазу үшін қажетті ақпараттар: басқарушы құрылымдар мен ауысулар есептері

Мақсаты:

- MIPS ассемблерінде шартты және шартсыз көшу командаларын қолдануды үйрену;
- beq, bne, slt командаларының жұмысын түсіну;
- Басқарушы құрылымдар арқылы (if, else, циклдер) қарапайым есептерді шығару.

Қысқаша теория:

Бағдарламалауда басқарушы құрылымдар негізгі үшке бөлінеді: сызықтық, тармақталу (if/else) және циклдік (for/while). Жоғары деңгейлі тілдерде if/else, for, while сияқты арнайы құрылымдар бар болса, төменгі деңгейлі ассемблер тілінде бұл құрылымдар шартты және шартсыз көшу командалары арқылы жүзеге асады.

MIPS архитектурасында шартсыз көшу j командасы арқылы орындалады. Бұл команда орындалғанда бағдарлама орындаушысы көрсетілген белгіге (label) бірден өтеді. Мысалы, j end деген команда бағдарлама орындалуын end белгісіне ауыстырады. Бұл жоғары деңгейлі тілдердегі goto операторына ұқсайды.

Шартты көшу командалары beq (branch if equal) және bne (branch if not equal). Бұл командалар екі регистрдің мәндерін салыстырады. Егер шарт орындалса, бағдарлама белгілі бір белгіге өтеді. beq \$t0, \$t1, label — егер \$t0 == \$t1 болса, label-ге көшу. Ал bne \$t0, \$t1, label — егер \$t0 != \$t1 болса, көшу орындалады.

Салыстыруда тек тең немесе тең емес жағдайлар емес, үлкен/кіші

жағдайлары да қажет болады. Ол үшін `slt` (`set on less than`) қолданылады. `slt $t2, $t0, $t1` командасы $\$t0 < \$t1$ болса, $\$t2 = 1$ етеді, әйтпесе 0. Кейін осы нәтижені шартты көшу командаларымен пайдалануға болады.

Циклдер де шартты көшу арқылы ұйымдастырылады. `For` немесе `while` циклі MIPS-та цикл басына `label` қойып, соңында шартты тексеріп, қажет болса қайтадан басына қайтудан тұрады. Осы арқылы қарапайым қайталанатын процестер жазылады.

Бұл зертханалық жұмыста студенттер шартты/шартсыз көшу командаларын қолданып, MIPS ассемблерінде басқарушы құрылымдарды құруды үйренеді. Бұл жоғары деңгейлі ойлаудан төменгі деңгейлі машиналық деңгейге көшу үшін маңызды қадам болып табылады.

Мысалдар (код үлгілері және түсіндірме):

Мысал 1. Шартсыз көшу (j)

```
.data
msg1: .asciiz "Бұл жол шықпайды\n"
msg2: .asciiz "Шартсыз көшу орындалды\n"

.text
main:
    j skip      # бірден skip белгісіне көшу
    li $v0, 4
    la $a0, msg1    # бұл орындалмайды
    syscall

skip:
    li $v0, 4
    la $a0, msg2    # орындалатын бөлік
    syscall
```

Түсіндірме: Бағдарлама бірден skip белгісіне өтеді, сондықтан бірінші хабарлама экранға шықпайды.

Мысал 2. Шартты көшу (beq, bne)

```
.data
msg1: .asciiz "Сандар тең емес\n"
msg2: .asciiz "Сандар тең\n"

.text
main:
    li $t0, 5
    li $t1, 5
    beq $t0, $t1, equal # егер $t0 == $t1 → equal
    li $v0, 4
    la $a0, msg1
    syscall
    j end

equal:
    li $v0, 4
    la $a0, msg2
    syscall

end:
    li $v0, 10
    syscall
```

Түсіндірме: Егер сандар тең болса, equal бөлігі орындалады. Әйтпесе алғашқы хабарлама шығады.

Мысал 3. Екі санның үлкенін табу (slt)

```
.text
main:
    li $t0, 12
    li $t1, 20
    slt $t2, $t0, $t1 # егер $t0 < $t1 → $t2 = 1
```

```
beq $t2, $zero, else # егер $t2 == 0 → else бөлігі
```

```
# if бөлігі: t0 < t1 → t1 үлкен
```

```
move $a0, $t1
```

```
li $v0, 1
```

```
syscall
```

```
j end
```

```
else:
```

```
move $a0, $t0
```

```
li $v0, 1
```

```
syscall
```

```
end:
```

```
li $v0, 10
```

```
syscall
```

Түсіндірме: Бағдарлама екі санды салыстырады және үлкенін экранға шығарады.

№3 зертханалық жұмыстың тапсырмалары.

1. Екі санның үлкенін экранға шығаратын бағдарлама жазыңыз.
2. Берілген сан тақ па, жұп па екенін анықтап, сәйкес хабарлама шығарыңыз.
3. Екі сан тең болса – «Сандар тең» хабарламасын шығаратын бағдарлама жасаңыз.
4. Шартсыз көшу командасын қолданып, экранға 1-ден 5-ке дейінгі сандарды шығарыңыз.
5. Сан теріс болса – «Теріс сан», оң болса – «Оң сан» хабарламасын шығаратын бағдарлама құрыңыз.