

Зертханалық жұмыс №4

Пән: Компьютер архитектурасы және операциялар келісімділігі

Тақырыбы: Ассемблер MIPS бойынша шартты және шартсыз операторлар қосымша есептерін алып шығару

Мақсаты:

- MIPS ассемблер тілінде күрделі тармақтану құрылымдарын (if, else if, else) құруды үйрену.
- Салыстыру командаларын (slt, slti, beq, bne) тиімді қолдану дағдысын қалыптастыру.
- Шартты өту командалары арқылы бағдарламаның орындау бағытын басқару.

Қысқаша теория

Күрделі шартты операторлар бірнеше мүмкін нәтиже жағдайын тексеру үшін қолданылады. Жоғары деңгейлі тілдердегі if - else if - else конструкциясы MIPS-те бірнеше салыстыру және шартты өту (branch) командаларының комбинациясы ретінде жасалады.

Негізгі командалар:

- **slt (set on less than):** slt \$t0, \$s0, \$s1 — егер \$s0 < \$s1 болса, \$t0 регистріне 1 мәнін, әйтпесе 0 мәнін жазеді.
- **slti (set on less than immediate):** slti \$t0, \$s0, 10 — егер \$s0 < 10 болса, \$t0 регистріне 1 мәнін, әйтпесе 0 мәнін жазеді.
- **beq (branch on equal):** beq \$s0, \$s1, label — егер \$s0 == \$s1 болса, label белгісіне өтеді.
- **bne (branch on not equal):** bne \$s0, \$s1, label — егер \$s0 != \$s1 болса, label белгісіне өтеді.
- **j (jump):** j label — шартсыз label белгісіне өтеді.

Жұмыс алгоритмі:

1. Бірінші шартты тексеру (if).
2. Егер шарт орындалса, сол тармақты орындап, соңына басқа тармақтарды өткізіп жіберу үшін j командасын қолдану.
3. Егер бірінші шарт орындалмаса, келесі шартты тексеру (else if).
4. Егер ешбір шарт орындалмаса, else тармағын орындау.

Мысал: Санды талдау (теріс, оң, нөл)

asm

.data

```
msg_negative: .asciiz "Сан теріс\n"
msg_positive: .asciiz "Сан оң\n"
msg_zero:     .asciiz "Сан нөлге тең\n"
```

```
.text
```

```
main:
```

```
    # Пайдаланушыдан сан енгізу
    li $v0, 5
    syscall
    move $t0, $v0 # Енгізілген сан $t0-да сақталады

    # 1-ші шарт: if (t0 < 0)
    slt $t1, $t0, $zero # if t0 < 0
    beq $t1, $zero, check_positive # if false, check positive
    # if body: теріс сан
    la $a0, msg_negative
    li $v0, 4
    syscall
    j exit      # remaining cases-ті өткізіп жіберу
```

```
check_positive:
```

```
    # 2-ші шарт: else if (t0 > 0)
    slt $t1, $zero, $t0 # if 0 < t0 (яғни t0 > 0)
    beq $t1, $zero, else # if false, it must be zero
    # else if body: оң сан
    la $a0, msg_positive
    li $v0, 4
    syscall
    j exit
```

```
else:
```

```
    # else body: нөл
    la $a0, msg_zero
    li $v0, 4
    syscall
```

```
exit:
```

```
    # Бағдарламаны аяқтау
    li $v0, 10
    syscall
```

Түсіндірме: Бұл бағдарлама пайдаланушыдан бүтін санды қабылдап, оның теріс, оң немесе нөлге тең екенін анықтап, сәйкес хабар шығарады.

Зертханалық жұмыс №4 тапсырмалары:

1. Пайдаланушыдан енгізілген санның жұп немесе тақ екенін анықтайтын бағдарлама жазыңыз. (Кеңес: `and` командасын 1-мен пайдаланып, ең аз маңызды битті тексеруге болады, немесе 2-ге бөліп, қалдықты тексеруге болады).
2. Пайдаланушыдан енгізілген үш санның ішіндегі ең үлкенін тауып шығаратын бағдарлама жазыңыз.
3. `$s0` регистріндегі бастапқы баллға қарай (0-100 аралығы) сәйкес әріптік бағаны (A, B, C, D, F) шығаратын бағдарлама құрыңыз. Мысалы, 90-дан жоғары - A, 80-ден жоғары - B, т.б.
4. Пайдаланушыдан енгізілген жылдың високос жыл ма, жоқ па екенін анықтайтын бағдарлама жазыңыз.
5. Пайдаланушыдан енгізілген символдың әріп (латынша), сан немесе арнайы таңба екенін анықтайтын бағдарлама жазыңыз.