

Procedure body

```
swap:  sll    $t1, $a1, 2           # reg $t1 = k * 4
      add    $t1, $a0, $t1         # reg $t1 = v + (k * 4)
                                           # reg $t1 has the address of v[k]
      lw     $t0, 0($t1)           # reg $t0 (temp) = v[k]
      lw     $t2, 4($t1)           # reg $t2 = v[k + 1]
                                           # refers to next element of v
      sw     $t2, 0($t1)           # v[k] = reg $t2
      sw     $t0, 4($t1)           # v[k+1] = reg $t0 (temp)
```

Procedure return

```
jr     $ra                          # return to calling routine
```

FIGURE 2.25 MIPS assembly code of the procedure *swap* in Figure 2.24.