

MIPS Assembly Cheat Sheet – Booklet

Πίνακας περιεχομένων

MIPS Assembly Cheat Sheet – Booklet.....	1
1. Registers.....	1
✚ Πίνακας Καταχωρητών.....	1
2. Memory Segments.....	2
3. System Calls.....	2
🔧 Βασικές εντολές.....	2
4. Arithmetic.....	2
✚ Πρόσθεση / Αφαίρεση.....	2
✖ Πολλαπλασιασμός / Διαίρεση.....	3
5. Bitwise / Shifts.....	3
🔄 Bitwise εντολές.....	3
🔄 Shift εντολές.....	3
6. Branches.....	3
🎯 Συνθήκες.....	3
7. Loops.....	4
8. Arrays / Memory.....	4
9. Functions.....	4
10. Stack Frames.....	4
11. Strings.....	4

1. Registers

✚ Πίνακας Καταχωρητών

- Συμβολικά ονόματα καταχωρητών:

\$0	\$2-\$3	\$4-\$7	\$8-\$15	\$16-\$23	\$24-\$25	\$28	\$29	\$30	\$31
\$zero	\$v0-\$v1	\$a0-\$a3	\$t0-\$t7	\$s0-\$s7	\$t8-\$t9	\$gp	\$sp	\$fp	\$ra

- Οι καταχωρητές \$1 (συμβολικό \$at) και \$26-\$27 (συμβολικά \$k0-\$k1) δεν είναι προσβάσιμοι στον προγραμματιστή, ενώ κάποιοι άλλοι επιτελούν ειδικές λειτουργίες
- Ο καταχωρητής \$0 (ή \$zero) έχει πάντοτε την τιμή 0

Κατηγορία	Registers	Χρήση
Σταθερός	\$zero	πάντα 0
Values	\$v0, \$v1	επιστροφή τιμών, system calls

Κατηγορία	Registers	Χρήση
Arguments	\$a0-\$a3	ορίσματα σε functions
Temporaries	\$t0-\$t9	προσωρινοί, δεν πρέπει να αποθηκεύονται
Saved	\$s0-\$s7	πρέπει να αποθηκεύονται/επαναφέρονται
Stack	\$sp	δείκτης στοίβας
Frame	\$fp	δείκτης frame
Return address	\$ra	jr επιστροφή

Οι βασικότερες κατηγορίες καταχωρητών και η χρήση τους.

Παράδειγμα:

```
li $t0,5
```

```
li $t1,7
```

```
add $t2,$t0,$t1
```

Άσκηση: Γράψε πρόγραμμα που προσθέτει τρεις αριθμούς και αποθηκεύει το αποτέλεσμα στο \$s0.

2. Memory Segments

.data για δεδομένα, .text για κώδικα.

Παράδειγμα:

```
.data msg: .asciiz "Hello" .text main: la $a0,msg
```

Άσκηση: Δήλωσε δύο μεταβλητές και φόρτωσέ τις σε καταχωρητές.

3. System Calls



Βασικές εντολές

\$v0	Λειτουργία	Παράμετρος
1	print int	\$a0 = number
4	print string	\$a0 = address
5	read int	επιστροφή → \$v0
8	read string	\$a0=buf, \$a1=len
10	exit	—

Χρήση του \$v0 για επιλογή λειτουργίας.

Παράδειγμα:

```
li $v0,4
```

```
la $a0,
```

```
msg syscall
```

Άσκηση: Ζήτα από τον χρήστη έναν αριθμό και εκτύπωσέ τον.

4. Arithmetic



Πρόσθεση / Αφαίρεση

```
add $t0,$t1,$t2
```

```
addi $t0,$t1,10
sub $t0,$t1,$t2
```

✖ Πολλαπλασιασμός / Διαίρεση

```
mul $t0,$t1,$t2
div $t1,$t2
mflo $t0 # quotient
mfhi $t0 # remainder
```

Πρόσθεση, αφαίρεση, πολλαπλασιασμός και διαίρεση.

Παράδειγμα:

```
li $t1,20
li $t2,3
div $t1,$t2
mflo $t3
mfhi $t4
```

Άσκηση: Υπολόγισε το $7 \cdot 9$ και αποθήκευσέ το στη μνήμη.

5. Bitwise / Shifts



Bitwise εντολές

```
and $t0,$t1,$t2
or $t0,$t1,$t2
xor $t0,$t1,$t2
nor $t0,$t1,$zero # NOT
```



Shift εντολές

```
sll $t0,$t1,2 # multiply by 4
srl $t0,$t1,2
```

Λογικές πράξεις και ολισθήσεις.

Παράδειγμα:

```
li $t1,5
sll $t0,$t1,1
```

Άσκηση: Κάνε shift-left έναν αριθμό κατά 3 θέσεις.

6. Branches



Συνθήκες

```
beq $t0,$t1,label
bne $t0,$t1,label
blt $t0,$t1,label
bgt $t0,$t1,label
```

```
slt $t0,$t1,$t2
```

Συνθήκες και διακλαδώσεις.

Παράδειγμα:

```
slt $t2,$t0,$t1
```

```
bne $t2,$zero,label
```

Άσκηση: Γράψε if-else που συγκρίνει δύο αριθμούς.

7. Loops

Επανάληψεις με branch και jump.

Παράδειγμα:

```
loop:
```

```
    bge $t0,$t1,
```

```
    end
```

```
loop:
```

```
    addi $t0,$t0,1 j
```

```
end:
```

Άσκηση: Υπολόγισε το άθροισμα 1..N.

8. Arrays / Memory

Προσπέλαση πίνακα με offset.

Παράδειγμα:

```
la $t0,arr lw $t1,0($t0)
```

Άσκηση: Φόρτωσε τον τρίτο αριθμό ενός πίνακα.

9. Functions

Κλήσεις με jal και επιστροφές με jr.

Παράδειγμα:

```
sum: add $v0,$a0,$a1 jr $ra
```

Άσκηση: Φτιάξε συνάρτηση που επιστρέφει max(a,b).

10. Stack Frames

Τυπικό πρότυπο αποθήκευσης καταχωρητών.

Παράδειγμα:

```
addi $sp,$sp,-8 sw $ra,4($sp) sw $s0,0($sp) ... lw $s0,0($sp) lw $ra,4($sp) addi $sp,$sp,8 jr $ra
```

Άσκηση: Φτιάξε συνάρτηση με δικές της τοπικές μεταβλητές.

11. Strings

Επεξεργασία byte-byte έως το null.

Παράδειγμα:

```
lb $t2,0($t0) beq $t2,$zero,end
```

Άσκηση: Υπολόγισε το μήκος μιας συμβολοσειράς.