

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ
ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 2

ΥΠΟΡΟΥΤΙΝΕΣ ΣΥΣΤΗΜΑΤΟΣ

Όπως έχει παρουσιαστεί στην προηγούμενη εργαστηριακή άσκηση, ο MIPS περιλαμβάνει υπορουτίνες συστήματος οι οποίες καλούνται θέτοντας τους καταχωρητές \$v0, \$a0 στις κατάλληλες τιμές. Συγκεκριμένα, όταν ο καταχωρητής \$v0 τεθεί στην τιμή 4, τότε η συμβολοσειρά που βρίσκεται στη διεύθυνση μνήμης που έχει αποθηκευτεί στον καταχωρητή \$a0 εκτυπώνεται αμέσως μετά την κλήση μέσω syscall. Γενικά, οι καταχωρητές που μπορούν να χρησιμοποιηθούν είναι οι **\$v0, \$a0, \$a1, \$a2, ή \$a3**.

Υπάρχουν 10 γνωστές υπορουτίνες συστήματος τις οποίες μπορείτε να χρησιμοποιήσετε για είσοδο/έξοδο δεδομένων. Παρακάτω ακολουθεί μια συνοπτική παρουσίαση:

Κωδικός υπορουτίνας (\$v0)	Υπορουτίνα	Σημαντικοί καταχωρητές (εκτός \$v0)	Κώδικας παραδείγματος
1	Εκτύπωση ακεραίου (Print an Integer)	ΕΙΣΟΔΟΣ \$a0 : περιέχει τον ακέραιο προς εκτύπωση	li \$a0, 12345 li \$v0, 1 syscall
2	Εκτύπωση κινητής υποδιαστολής (Print a Float)	ΕΙΣΟΔΟΣ \$f12 : περιέχει τον αριθμό κινητής υποδιαστολής προς εκτύπωση	.data pin: .float 3.1413 .text l.s \$f12, pin li \$v0, 2 syscall
3	Εκτύπωση κινητής υποδιαστολής διπλής ακρίβειας (Print a Double)	ΕΙΣΟΔΟΣ \$f12 : περιέχει τον αριθμό κινητής υποδιαστολής διπλής ακρίβειας προς εκτύπωση	.data ddn: .double 1.2345678901234567 .text l.d \$f12, ddn li \$v0, 3 syscall

4	Εκτύπωση συμβολοσειράς (Print a String)	ΕΙΣΟΔΟΣ \$a0 : δείκτης στη διεύθυνση μνήμης που περιέχει τη συμβολοσειρά προς εκτύπωση	.data
			str: .asciiz "This is an example!"
5	Ανάγνωση ακεραίου (Read an Integer)	ΕΞΟΔΟΣ \$v0 : περιέχει τον ακέραιο αριθμό που διαβάστηκε από το πληκτρολόγιο	.text
			la \$a0, str
6	Ανάγνωση κινητής υποδιαστολής (Read a Float)	ΕΞΟΔΟΣ \$f0 : περιέχει τον αριθμό κινητής υποδιαστολής που διαβάστηκε από το πληκτρολόγιο	li \$v0, 4
			syscall
7	Ανάγνωση κινητής υποδιαστολής διπλής ακρίβειας (Read a double)	ΕΞΟΔΟΣ \$f0 : περιέχει τον αριθμό κινητής υποδιαστολής διπλής ακρίβειας που διαβάστηκε από το πληκτρολόγιο	li \$v0, 5
			syscall
8	Ανάγνωση συμβολοσειράς (Read a String)	ΕΙΣΟΔΟΣ \$a0 : Η διεύθυνση του μνήμης εισόδου (input buffer) \$a1 : χωρητικότητα της μνήμης εισόδου (σε bytes)	.data
			str: .space 255
9	Δυναμική δέσμευση n bytes μνήμης (Dynamically allocate n bytes of memory)	ΕΞΟΔΟΣ Η μνήμη εισόδου περιέχει την συμβολοσειρά που διαβάστηκε από το πληκτρολόγιο	.text
			la \$a0, str
10	Ανάγνωση συμβολοσειράς (Read a String)	ΕΙΣΟΔΟΣ \$a0 : Η διεύθυνση του μνήμης εισόδου (input buffer) \$a1 : χωρητικότητα της μνήμης εισόδου (σε bytes)	li \$a1, 255
			li \$v0, 8
11	Ανάγνωση συμβολοσειράς (Read a String)	ΕΞΟΔΟΣ Η μνήμη εισόδου περιέχει την συμβολοσειρά που διαβάστηκε από το πληκτρολόγιο	syscall
			li \$v0, 4
12	Ανάγνωση συμβολοσειράς (Read a String)	ΕΙΣΟΔΟΣ \$a0 : Η διεύθυνση του μνήμης εισόδου (input buffer) \$a1 : χωρητικότητα της μνήμης εισόδου (σε bytes)	syscall
			li \$v0, 4
13	Ανάγνωση συμβολοσειράς (Read a String)	ΕΙΣΟΔΟΣ \$a0 : Η διεύθυνση του μνήμης εισόδου (input buffer) \$a1 : χωρητικότητα της μνήμης εισόδου (σε bytes)	li \$a0, 255
			# Set n bytes to allocate
14	Ανάγνωση συμβολοσειράς (Read a String)	ΕΙΣΟΔΟΣ \$a0 : n Το πλήθος των bytes προς δέσμευση	li \$v0, 9
			syscall
15	Ανάγνωση συμβολοσειράς (Read a String)	ΕΞΟΔΟΣ \$v0 : Δείκτης στη διεύθυνση που περιέχει τα bytes που δεσμεύτηκαν	# Allocate mem
			move \$a0, \$v0
16	Ανάγνωση συμβολοσειράς (Read a String)	ΕΙΣΟΔΟΣ \$a0 : Η διεύθυνση του μνήμης εισόδου (input buffer) \$a1 : χωρητικότητα της μνήμης εισόδου (σε bytes)	li \$a1, 255
			li \$v0, 8

		syscall # Read string from input
		li \$v0, 4
		syscall # Print string to output
10	Έξοδος (Exit)	li \$v0, 10
		syscall

Έχοντας τα παραπάνω ως πληροφορία, να παραδώσετε την αναφορά όπως έχει ανακοινωθεί στο e-class. Η αναφορά θα πρέπει να απαντά στα παρακάτω:

ΕΡΓΑΣΙΑ 1

Ένα πρόγραμμα το οποίο θα διαβάζει από το πληκτρολόγιο το επίθετό σας και το ΑΜ (ως ακέραιο αριθμό) και στη συνέχεια θα τα εκτυπώνει στην οθόνη (console).

ΕΡΓΑΣΙΑ 2

Να δημιουργήσετε ένα πρόγραμμα το οποίο θα διαβάζει από το πληκτρολόγιο 2 ακέραιους αριθμούς και στη συνέχεια θα εκτυπώνει τα παρακάτω μηνύματα:

The two numbers <are/are not> equal

The greater number is number <1>

The smaller number is number <2>

Όπου <> αντικαταστήστε με το κατάλληλο.

Παράδειγμα 1

```

2 8
The two numbers are not equal
The greater number is number 8
The smaller number is number 2

```

Παράδειγμα 2

```

4 4
The two numbers are equal
The greater number is number 4
The smaller number is number 4

```

ΣΗΜΕΙΩΣΗ

```

li $v0, 1                    # Load immediate $v0 with value 1

bgtz $v0, $FLOW1           # Branch to $FLOW1 label, if $v0 > 0

bge $s1, $s0, L1           # Branch to L1 label, if ($s1 ≥ $s0)

j L2                        # Branch to L2 label, unconditionally

beq $s0, $s1, L3           # Branch to L1 label, if ($s1 = $s0)

```