



1η ΑΣΚΗΣΗ ΣΤΗΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ

Ακ. έτος 2023-2024, 5ο Εξάμηνο, Σχολή ΗΜ&ΜΥ

Τελική Ημερομηνία Παράδοσης: **26/11/2023**

ΜΕΡΟΣ Α

Δίνεται το παρακάτω πρόγραμμα γραμμένο σε C, καθώς και η αντίστοιχη μετάφραση του σε assembly MIPS. Συμπληρώστε τα κενά. Σας υπενθυμίζουμε ότι ο καταχωρητής \$zero είναι πάντα μηδέν.

	fib:	addi	\$sp	,	\$sp	,	_____
		sw	\$ra	,	_____		
		sw	\$s0	,	_____		
		_____	\$t0	,	_____	,	2
		_____	\$t0	,	_____	,	else
int fib(int n) {		addi	_____	,	\$zero	,	_____
if (n < 2)		addi	_____	,	_____	,	12
return 1;		_____	\$ra				
else		addi	\$a0	,	\$a0	,	_____
return fib(n-1)+fib(n-2);	else:	sw	_____	,	_____		
}		_____	fib				
		add	\$s0	,	_____	,	\$zero
		lw	_____	,	_____		
		addi	\$a0	,	\$a0	,	_____
		_____	fib				
		add	\$v0	,	_____	,	\$s0
		lw	_____	,	_____		
		lw	_____	,	_____		
		addi	\$sp	,	\$sp	,	_____
		jr	_____				

ΜΕΡΟΣ Β

(i) Υλοποιήστε σε assembly MIPS τη διαδικασία **encode_caesar_ece**, η οποία χρησιμοποιεί τον «Κώδικα του Καίσαρα» (Caesar cipher¹) για την κωδικοποίηση ενός μηνύματος κειμένου αντικαθιστώντας τα γράμματα με άλλα που βρίσκονται σε σταθερή απόσταση στην αλφάβητο από αυτά. Θεωρήστε τα εξής:

- Το μήνυμα κειμένου είναι μια συμβολοσειρά (null-terminated ASCII string), η οποία περιέχει μόνο κεφαλαία γράμματα του λατινικού αλφαβήτου και τον χαρακτήρα κενό (space).
- Ο χαρακτήρας κενό (ASCII 0x20) και ο χαρακτήρας NULL (ASCII 0x00) δεν κωδικοποιούνται.
- Η κωδικοποίηση του μηνύματος γίνεται in-place, δηλαδή το κωδικοποιημένο μήνυμα γράφεται στις θέσεις μνήμης που καταλαμβάνει το αρχικό μήνυμα.

¹ https://en.wikipedia.org/wiki/Caesar_cipher

- Κάθε γράμμα αντικαθίσταται από το γράμμα που βρίσκεται n θέσεις πιο πριν στο λατινικό αλφάβητο, όπου n θετικός ακέραιος, για τον οποίο ισχύει $1 \leq n \leq 25$. Για παράδειγμα, αν $n = 3$, το γράμμα D θα αντικατασταθεί από το A και το B από το Y.
- Η διαδικασία υπολογίζει το n με βάση τον αριθμό μητρώου (AM) της σχολής. Συγκεκριμένα, το υπολογίζει σύμφωνα με τον τύπο $n = AM \% 25 + 1$.

Η διαδικασία που θα υλοποιήσετε θα δέχεται δύο ορίσματα:

- Στον \$a0 ένα δείκτη στη θέση του πρώτου στοιχείου του κειμένου προς κωδικοποίηση.
- Στον \$a1 τον αριθμό μητρώου.

Παράδειγμα:

Στη διεύθυνση 0x1000 είναι αποθηκευμένο σε ASCII κωδικοποίηση το μήνυμα “HELLO CSLAB”:

Διεύθυνση	Bytes							
0x1000	0x48	0x45	0x4C	0x4C	0x4F	0x20	0x43	0x53
0x1008	0x4C	0x41	0x42	0x00	0x00	0x00	0x00	0x00

Μετά την κλήση της διαδικασίας encode_caesar_ece με ορίσματα τη διεύθυνση 0x1000 και τον AM 3119891, η μνήμη θα περιέχει τα παρακάτω δεδομένα, τα οποία αντιστοιχούν στο μήνυμα “QNUUX LBUJK” που προκύπτει από το αρχικό μήνυμα για $n = 17$.

Διεύθυνση	Bytes							
0x1000	0x51	0x4E	0x55	0x55	0x58	0x20	0x4C	0x42
0x1008	0x55	0x4A	0x4B	0x00	0x00	0x00	0x00	0x00

(ii) Εκτελέστε τη διαδικασία που υλοποιήσατε με ορίσματα το ονοματεπώνυμο και το AM σας και δώστε το κωδικοποιημένο μήνυμα που σας επέστρεψε.

ΜΕΡΟΣ Γ

Υλοποιήστε μια διαδικασία σε assembly MIPS η οποία υπολογίζει αναδρομικά την τιμή της συνάρτησης Ackermann² για δύο μη αρνητικούς αριθμούς σύμφωνα με τον παρακάτω τύπο:

$$A(x, y) = \begin{cases} y + 1 & , \text{όταν } x = 0 \\ A(x - 1, 1) & , \text{όταν } y = 0 \\ A(x - 1, A(x, y - 1)) & , \text{σε κάθε άλλη περίπτωση} \end{cases}$$

Η διαδικασία θα πρέπει να δέχεται ως είσοδο δύο ακέραιους μη αρνητικούς αριθμούς, να ελέγχει για σωστή είσοδο δεδομένων και σε περίπτωση αρνητικής τιμής να επιστρέφει -1. Εάν οι είσοδοι είναι σωστές, η διαδικασία θα πρέπει να επιστρέφει την τιμή της συνάρτησης Ackermann για τους συγκεκριμένους αριθμούς.

² https://en.wikipedia.org/wiki/Ackermann_function

Για την υλοποίηση της άσκησης μπορείτε να χρησιμοποιήσετε κάποιον MIPS emulator, όπως ο **SPIM**³ ή ο **MILE**⁴ που αναπτύχθηκε από συμφοιτητές σας και διατίθεται από το Εργαστήριο Υπολογιστικών Συστημάτων. Στα εργαλεία αυτά μπορείτε να γράφετε MIPS assembly και να την εκτελείτε παρακολουθώντας τα περιεχόμενα των καταχωρητών και της μνήμης καθιστώντας έτσι ευκολότερη την παραγωγή και τον έλεγχο του απαιτούμενου κώδικα.

Παραδοτέο της άσκησης θα είναι ένα ηλεκτρονικό κείμενο (**pdf, docx ή odt**) που θα περιέχει τους **κώδικες assembly και των 3 μερών** της άσκησης. Ο κώδικας θα πρέπει να περιέχει αναλυτικά σχόλια για την κατανόηση της λύσης σας από τους διδάσκοντες.

Στο ηλεκτρονικό κείμενο να αναφέρετε στην αρχή τα στοιχεία σας (Όνομα, Επώνυμο, ΑΜ).

Η άσκηση θα παραδοθεί ηλεκτρονικά στο moodle του μαθήματος:

<https://helios.ntua.gr/course/view.php?id=1038>

Δουλέψτε ατομικά. Έχει ιδιαίτερη αξία για την κατανόηση του μαθήματος να κάνετε μόνοι σας την εργασία. Μην προσπαθήσετε να την αντιγράψετε από άλλους συμφοιτητές σας.

³ <https://spimsimulator.sourceforge.net/>

⁴ <http://www.cslab.ece.ntua.gr/courses/comparch/assign.go>