

Οικονομικό Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής
Μάθημα: **Οργάνωση Συστημάτων Υπολογιστών**
Χειμερινό Εξάμηνο 2017-18
Διδάσκων: Αν. Καθηγητής Ι. Κουτσόπουλος

Υπεύθυνη Εργασιών:

Μαρία Τογαντζή

Μέλος ΕΔΙΠ

e-mail: mst@aueb.gr

Ημερομηνία δημοσίευσης: 13/12/2017

Προθεσμία υποβολής: 12/01/2018

3η Εργασία Προγραμματισμού σε Assembly

Εισαγωγή: Στον επεξεργαστή MIPS32, η δυναμική παραχώρηση μνήμης, γίνεται με την κλήση 9 του συστήματος η οποία επιστρέφει ένα δείκτη προς την περιοχή της μνήμης που παραχωρεί το σύστημα. Το μέγεθος σε bytes της περιοχής μνήμης που παραχωρείται, δίνεται ως τιμή στον καταχωρητή \$a0.

sbrk - Memory Allocation:

- system call 9,
- \$a0 = number of bytes of storage desired,
- \$v0 = address of block

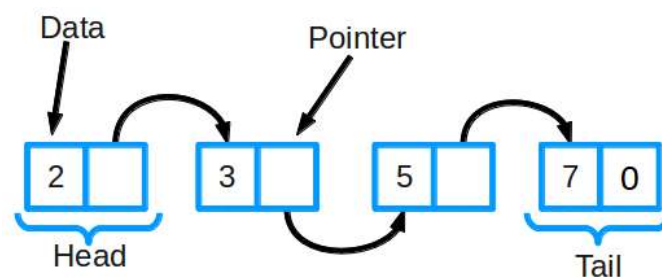
```

1      .text
2
3      .globl main
4  main:
5
6      li  $a0,8          # 2 x sizeof(int)
7      li  $v0,9          # sbrk - system call 9, κλήση υπηρεσίας για
8      syscall           # την παραχώρηση 2 λέξεων μνήμης (8 bytes)
9
10     move $t1,$v0       # $t1 = διεύθυνση πρώτης λέξης στη μνήμη
11
12     li  $v0,5          # read (int1)
13     syscall
14     sw  $v0,($t1)      # store int1 στην πρώτη θέση
15
16     li  $v0,5          # read (int2)
17     syscall
18     sw  $v0,4($t1)     # store int2 στη δεύτερη θέση
19
20     lw  $a0,($t1)      # print (int1)
21     li  $v0,1
22     syscall
23
24     lw  $a0,4($t1)     # print (int2)
25     li  $v0,1
26     syscall
27
28     li  $v0,10         # exit
29     syscall

```

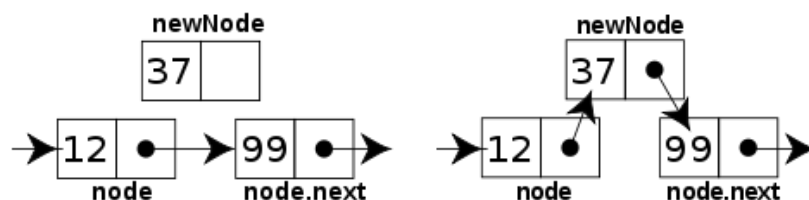
Για παράδειγμα, το παραπάνω πρόγραμμα, διαβάζει δύο ακραίους και τους αποθηκεύει στη μνήμη, σε περιοχή που παραχωρείται δυναμικά, κατά τη διάρκεια εκτέλεσης του προγράμματος, χωρίς δήλωση και αποθήκευση σε πίνακα ακεραίων στο τμήμα δεδομένων του προγράμματος. Τέλος, εμφανίζει τους δύο ακραίους.

Περιγραφή εργασίας: Στην περίπτωση μιας γραμμικής συνδεδεμένης λίστας, κάθε κόμβος της θα μπορούσε να είναι μια περιοχή της μνήμης που παραχωρείτε με τον παραπάνω τρόπο και αποτελείται από δύο ακραίους. Ο πρώτος αποτελεί το δεδομένο (data) του κόμβου και ο δεύτερος το σύνδεσμο (link) προς τον επόμενο κόμβο. Ο τελευταίος κόμβος θα μπορούσε να έχει ως link το μηδέν:

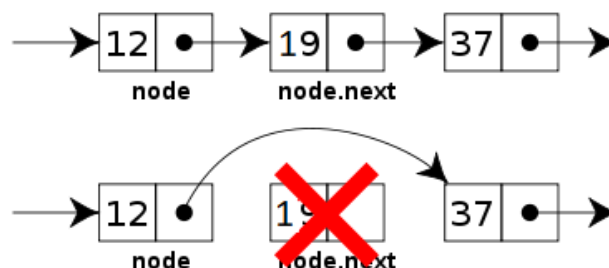


Γράψτε ένα πρόγραμμα στη συμβολική γλώσσα του επεξεργαστή MIPS32, που δημιουργεί και διαχειρίζεται μια δυναμική λίστα ακεραίων. Κάθε ακέραιος πρέπει να αποθηκεύεται σε ένα κόμβο της λίστας. Η εισαγωγή ή η διαγραφή κόμβου στη λίστα θα πρέπει να γίνεται με τέτοιο τρόπο ώστε η λίστα να παραμένει πάντα ταξινομημένη σε αύξουσα σειρά:

Εισαγωγή Κόμβου στη λίστα:



Διαγραφή Κόμβου από τη λίστα:



Η διεπαφή με το χρήστη, πρέπει να γίνεται μέσω **μιας λίστας επιλογών (menu)** που θα παρέχει στο χρήστη τις εξής επιλογές:

1. Εισαγωγή Κόμβου στη Λίστα στη σωστή θέση, έτσι ώστε η λίστα να είναι ταξινομημένη σε αύξουσα σειρά.
2. Διαγραφή Κόμβου από τη λίστα, έτσι ώστε η λίστα να παραμένει ταξινομημένη σε αύξουσα σειρά.
3. Εμφάνιση Λίστας με αύξουσα σειρά (προβολή δεδομένων από τον πρώτο ως τον τελευταίο κόμβο).
4. Εμφάνιση Λίστας με φθίνουσα σειρά (προβολή δεδομένων από τον τελευταίο κόμβο ως τον πρώτο κόμβο).

Παρατηρήσεις σχετικά με την υλοποίηση:

- Η υλοποίηση πρέπει να γίνει όπως περιγράφεται παραπάνω μέσω κλήσεων δυναμικής παραχώρησης μνήμης (**sbrk**) και όχι με πίνακα ακεραίων που έχει δηλωθεί στο data segment.
- Όλες οι επιλογές του μενού θα πρέπει να γίνονται μέσω υποπρογραμμάτων. Γενικότερα σχεδιάστε το πρόγραμμα με τέτοιο τρόπο ώστε να μεγιστοποιείται η επαναχρησιμοποίηση του κώδικα μέσω υποπρογραμμάτων.
- Το υποπρόγραμμα που υλοποιεί την επιλογή (4) πρέπει να είναι **αναδρομικό (recursive)**.

Οδηγίες Παράδοσης:

- Η εργασία είναι ομαδική **2 ατόμων**.
- **Ονομάστε το αρχείο** που περιέχει το πρόγραμμα, με τους αριθμούς του φοιτητικού σας μητρώου και κατάληξη **.s** (για παράδειγμα 3160000_3160010.s).
- Ακολουθείστε τις παρακάτω **οδηγίες** σχετικά με τον τρόπο συγγραφής του προγράμματος και την τεκμηρίωσή του:

α. Στην αρχή του προγράμματος πρέπει να υπάρχει σχόλιο με τις παρακάτω πληροφορίες:

```
# Authors:
```

```
# Date:
```

β. Πριν από κάθε **υποπρόγραμμα** περιγράψτε με τη μορφή ψευδοκώδικα το υποπρόγραμμα που υλοποιείτε. Περιγράψτε την επικεφαλίδα του υποπρογράμματος, με τις παραμέτρους και τα επιστρεφόμενα αποτελέσματα καθώς και τον αλγόριθμο που υλοποιεί το υποπρόγραμμα.

```
# Pseudocode:
```

```
#
```

```
#
```

γ. Περιλάβετε επίσης σύντομα **σχόλια σε επιμέρους εντολές ή ομάδες εντολών**, για παράδειγμα:

```
printQuestion:
    li    $v0, 4      # print(quest)
    la    $a0, quest
    syscall
```

- Εκτελέστε το πρόγραμμά σας στον προσομοιωτή **SPIM** και βεβαιωθείτε ότι δεν έχει συντακτικά ή άλλα σφάλματα.
- **Συμπιέστε** (.zip ή .rar) το αρχείο που περιέχει το πρόγραμμα και **ανεβάστε** το στο eclass μέχρι την αναφερόμενη ημερομηνία.