



## Γλώσσες Προγραμματισμού II

Οι ασκήσεις πρέπει να παραδοθούν στους διδάσκοντες σε ηλεκτρονική μορφή μέσω του συνεργατικού συστήματος ηλεκτρονικής μάθησης [moodle.softlab.ntua.gr](http://moodle.softlab.ntua.gr). Η προθεσμία παράδοσης θα τηρείται αυστηρά. Έχετε δικαίωμα να καθυστερήσετε το πολύ μία άσκηση.

### Άσκηση 9 Δυναμική διαχείριση μνήμης μέσω συλλογής σκουπιδιών

Προθεσμία παράδοσης: 7/4/2013

Στη διάλεξη του μαθήματος μιλήσαμε για διάφορους αλγόριθμους δυναμικής διαχείρισης μνήμης και αναφέραμε και την έννοια της συντηρητικής συλλογής σκουπιδιών (conservative garbage collection). Ο πιο διαδεδομένος τέτοιος συλλέκτης σκουπιδιών είναι αυτός των Boehm-Demers-Weiser. Η σελίδα [http://www.hp1.hp.com/personal/Hans\\_Boehm/gc](http://www.hp1.hp.com/personal/Hans_Boehm/gc) έχει πολλές πληροφορίες για αυτόν, καθώς και τα slides από ένα tutorial που ο συγκεκριμένος ερευνητής έκανε το 2004 σε ένα συνέδριο:

[http://www.hp1.hp.com/personal/Hans\\_Boehm/gc/04tutorial.pdf](http://www.hp1.hp.com/personal/Hans_Boehm/gc/04tutorial.pdf)

Διαβάστε τα slides, πιθανά βρείτε ότι άλλες πληροφορίες σας χρειάζονται, και απαντήστε στις παρακάτω δέκα ερωτήσεις:

1. (Slide 6: "What else can it do?") "But objects are not moved": Πώς ακριβώς λοιπόν γίνεται η συλλογή σκουπιδιών ανά γενεές χωρίς να μετακινούνται τα αντικείμενα;
2. (Slide 7: "What else can it do?") "And synchronous heap finalizers are broken anyway...": Τι είναι οι σύγχρονοι finalizers και γιατί ο συγγραφέας ισχυρίζεται ότι 'είναι σπασμένοι έτσι και αλλιώς';
3. (Slide 9: "C Interface overview") "Libunwind, addr2line allow better alternatives": Πώς και γιατί;
4. (Slide "Asymptotic Complexity of Mark-Sweep vs. Copying") Συζητήστε το επιχείρημα που χρησιμοποιεί ο συγγραφέας και το κατά πόσο συμφωνείτε ή διαφωνείτε μαζί του.
5. (Slide "Locating roots") "Very platform dependent": Διαλέξτε την αγαπημένη σας πλατφόρμα (π.χ. x86, x86\_64), βρείτε πώς υλοποιείται στον κώδικα του συλλέκτη και περιγράψτε το συνοπτικά.
6. (Slides "Generational, Incremental, Mostly Concurrent GC" και "Mostly Concurrent GC") Περιγράψτε συνοπτικά, όπως θα το εξηγούσατε σε έναν συμφοιτητή σας, πώς δουλεύει ο mostly concurrent garbage collector.
7. (Slide "Finalization (quick observation)") Εξηγήστε τα θέματα που περιγράφονται μετά από την πρόταση "Finalizers *must* introduce concurrency ..." μέχρι το τέλος του slide.
8. (Slide "Conclusions" πριν από το "αρχικό" slide με τον τίτλο "Bounding ...") "There may be a cache cost ... But I do not think we fully understand this yet": Η συγκεκριμένη παράγραφος είναι από το 2004. Ψάξτε τη σχετική βιβλιογραφία για να βρείτε αν έχει βελτιωθεί κάπως η κατάσταση από τότε σε σχέση με την κατανόηση του συγκεκριμένου θέματος και περιγράψτε τα ευρήματά σας.
9. (Slide "GC robustness") Ο συγγραφέας ορίζει την έννοια του weakly GC robust program. Τι γνώμη έχετε για αυτήν;
10. (Slide "Conclusions") Περιγράψτε συνοπτικά τα επιπλέον θέματα που προκύπτουν σε μια γλώσσα με σκληρή αποτίμηση.