

HY586 - Κατανεμημένος Υπολογισμός
Χειμερινό Εξάμηνο Ακ. Έτους 2009-10
Διδάσκουσα: Παναγιώτα Φατούρου

Θεωρητικές Εργασίες

Προθεσμία Παράδοσης: Αναφορά: 18 Ιανουαρίου 2010
Παρουσίαση: 22 Ιανουαρίου 2010

Ανάθεση Projects

Εργασία 1 (Universal Constructions) [Γιάννης Μακρυδάκης]

1. Afek, Dauber and Touitou, "Wait-Free Made Fast", *Proceedings of the ACM Symposium on Distributed Computing*, 1995 (<http://portal.acm.org/citation.cfm?id=225271>)
2. J. Anderson and M. Moir, "Universal Constructions for Large Objects", *IEEE Transactions on Parallel and Distributed Systems*, 10(12):1317-1332, December 1999 (contact the instructor)
3. P. Fatourou and N. Kallimanis, "The Red-Blue Adaptive Universal Constructions", *Proceedings of the International Symposium on Distributed Computing (DISC)*, 2009 (contact the instructor)

Εργασία 2 (Adaptive Algorithms) [Μάρκος Φουντουλάκης]

1. H. Attiya and A. Fouren, "Adaptive and Efficient Algorithms for Lattice Agreement and Renaming", *SIAM Journal on Computing*, 31(2): 642-664, 2001 (contact the instructor)
2. Hagit Attiya, Arie Fouren, "Algorithms adapting to point contention", *Journal of the ACM*, 50(4): 444-468 (2003) (<http://portal.acm.org/citation.cfm?id=792541>)
3. Hagit Attiya, Arie Fouren and Eli Gafni, "A Polynomial Adaptive Algorithm for Long-Lived $(2k-1)$ -Renaming", *Proceedings of the International Symposium on Distributed Computing*, 2000 (full version: <http://www.cs.technion.ac.il/~hagit/publications/AFG03.ps>)

Εργασία 3 (Snapshots) [Γιάννης Κλωνάτος]

1. Hagit Attiya, Ophir Rachman, "Atomic Snapshots in $O(n \log n)$ Operations", *SIAM Journal on Computing*, 27(2): 319-340 (1998) (contact the instructor)
2. Hagit Attiya, Arie Fouren, "Adaptive Wait-Free Algorithms for Lattice Agreement and Renaming", *Proceedings of the Annual ACM symposium on Principles of Distributed Computing (PODC)*, 1998: 277-286 (<http://portal.acm.org/citation.cfm?id=277749>)
3. Y. Afek, G. Stupp and D. Touitou, "Long-lived Adaptive Collect with Applications", *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, 1999 (<http://portal.acm.org/citation.cfm?id=796475>)

Εργασία 4 (Distributed Data Structures – Memory Reclamation) [Μιχάλης Αλβανός]

1. M. Michael. "High Performance Dynamic Lock-Free Hash Tables and List-Based Sets", In *Proceedings of the 14th annual ACM Symposium on Parallel algorithms and architectures*

(SPAA), pages 73-82, August 2002 (<http://www.research.ibm.com/people/m/michael/spaa-2002.pdf>)

2. M. Fomitchев and E. Ruppert. "Lock-Free Linked Lists and Skip Lists", In *Proceedings of the 23rd annual ACM symposium on Principles of distributed computing (PODC)*, pages 50-59, July 2004 (<http://www.cse.yorku.ca/~ruppert/papers/lfl.pdf>)
3. Maged M. Michael: Hazard Pointers: Safe Memory Reclamation for Lock-Free Objects. *IEEE Transactions on Parallel and Distributed Systems*, 15(6): 491-504 (2004) (contact the instructor)

Περιγραφή Project

Κάθε θεωρητικό project αποτελείται από τη μελέτη και κατανόηση τριών εργασιών (papers), τη συγγραφή ενός παραδοτέου όπου θα περιγράφονται οι εργασίες που μελετήσατε και την παρουσίαση μέρος των εργασιών αυτών στην τάξη. Το τεχνικό κομμάτι της παρουσίασης θα πρέπει να δομηθεί κατάλληλα, βάσει της συνοχής των αποτελεσμάτων που παρουσιάζονται στις διάφορες εργασίες, τις σχέσεις μεταξύ τους και των βελτιώσεων που υπάρχουν από εργασία σε εργασία.

Κατανόηση Εργασιών: Θα πρέπει να επιτευχθεί σε βάθος κατανόηση των εργασιών που έχετε αναλάβει. Πιο συγκεκριμένα θα πρέπει:

- ο να γνωρίζετε τους αλγορίθμους και τις τεχνικές που παρουσιάζονται σε αυτές,
- ο να μπορείτε να απαντάτε σε ερωτήματα του στυλ «Γιατί είναι χρήσιμη κάθε γραμμή των αλγορίθμων που μελετάτε και τι κακό μπορεί να συμβεί αν οποιαδήποτε γραμμή κώδικα δεν υπήρχε»,
- ο να επενδύσετε χρόνο στους αλγορίθμους που μελετάτε, να επινοήσετε τα δικά σας άσχημα σενάρια εκτέλεσης και να κατανοήσετε πως οι αλγόριθμοι αντεπεξέρχονται σε αυτά,
- ο να μελετήσετε μεγάλο αριθμό παραδειγμάτων που θα εξηγούν τη λειτουργία των αλγορίθμων που μελετάτε και να ανιχνεύσετε τα πιο δύσκολα σενάρια που καλούνται αυτοί να αντιμετωπίσουν,
- ο να φτιάξετε μεγάλο αριθμό παραδειγμάτων (επιπρόσθετα όσων ενδεχομένως παρουσιάζονται στην εργασία που μελετάτε) που να αποδεικνύουν ότι έχετε μελετήσει σε βάθος τους αλγορίθμους των εργασιών σας,
- ο να επιμείνετε στην ανάλυση των αλγορίθμων και να κατανοήσετε τις αποδείξεις που περιγράφονται στις εργασίες που έχετε αναλάβει,
- ο να ξαναγράψετε πιο αναλυτικά αποδείξεις των οποίων η κατανόηση σας δυσκόλεψε.

Τέλος, στην αναφορά σας θα πρέπει να παρουσιάσετε για κάθε αλγόριθμο καθώς και για την ανάλυσή του, μια διαισθητική περιγραφή του αλγορίθμου, της ορθότητάς του και της πολυπλοκότητάς του.

Συγγραφή Αναφοράς

Η αναφορά σας δεν θα πρέπει να ξεπερνά τις 20 (ενδεχόμενα πυκνογραμμένες) σελίδες. Επιπλέον σελίδες θα αξιολογούνται αρνητικά. Στην αναφορά σας πρέπει να συμπεριλάβετε τα σημαντικότερα κατά τη γνώμη σας αποτελέσματα των εργασιών που μελετάτε. Επίσης, θα πρέπει να παρουσιάσετε πολλά παραδείγματα που θα αποδεικνύουν την εις βάθος κατανόηση του θέματός σας.

Η αναφορά σας θα πρέπει να έχει περίπου της εξής δομή:

1. Εισαγωγή: Αφηρημένη περιγραφή του προβλήματος και της σημαντικότητας του. Αφηρημένη περιγραφή των αποτελεσμάτων που θα παρουσιαστούν στη συνέχεια (περίπου 1-2 σελίδα). Μια παράγραφος για την οργάνωση της υπόλοιπης αναφοράς.
2. Μοντέλο: Παρουσίαση του μοντέλου (σε τι σύστημα θα δουλέψετε, ορισμούς που χρειάζεστε για την περιγραφή και την ανάλυση του αλγορίθμου, κλπ.) (περίπου 2 σελίδες).
3. Τεχνικό μέρος της εργασίας: Εδώ θα περιγραφούν οι αλγόριθμοι, η ορθότητά τους και η ανάλυσή τους σύμφωνα με τις οδηγίες που σας δόθηκαν πιο πάνω (περίπου 14 σελίδες).
4. Επίλογος: Περιγραφή ανοιχτών προβλημάτων (αν υπάρχουν) (1 σελίδα).

Η αναφορά σας θα βαθμολογηθεί βάσει των εξής κριτηρίων:

1. πόσο πείθει ότι έχετε καταλάβει σε βάθος τα αποτελέσματα που παρουσιάζονται στις εργασίες που μελετήσατε,
2. πόσο καλά δομημένη είναι,
3. επιλογή αποτελεσμάτων που αποφασίσατε να συμπεριλάβετε (δεν πρέπει να είναι π.χ., μόνο τα πιο εύκολα αποτελέσματα που παρουσιάζονται στις εργασίες),
4. πόσο διαφοροποιείται από μια απλή μετάφραση των εργασιών,
5. πόσα (δικά σας) παραδείγματα έχετε μελετήσει και πόσο καλά εξηγείτε τους αλγορίθμους, τις τεχνικές και τις αποδείξεις που έχετε συμπεριλάβει,
6. ορθότητα διαισθητικών περιγραφών που έχετε προσθέσει,
7. πόσο καλογραμμένη είναι (βαθμός φορμαλισμού, πόσο σωστά είναι αυτά που γράφετε),
8. σε ποιο βαθμό επιτυγχάνονται όλοι οι παραπάνω στόχοι χωρίς να καταστρατηγείται το όριο των 12 σελίδων.

Παρουσίαση

Κάθε παρουσίαση θα πρέπει να διαρκεί μισή ώρα. Σιγουρευτείτε ότι η παρουσίασή σας δεν απαιτεί περισσότερο χρόνο (γιατί θα βαθμολογηθείτε βάσει όσων θα προλάβετε να παρουσιάσετε σε μισή ώρα). Κάθε παρουσίαση πρέπει να έχει περίπου την εξής μορφή:

1. Περιγραφή του προβλήματος που μελετάται.
2. Σύντομη περιγραφή των αποτελεσμάτων που θα παρουσιαστούν.
3. Περιγραφή Αλγορίθμων (κύρια με διαισθητικό τρόπο).
4. Πολλά παραδείγματα για την καλή κατανόηση του τρόπου λειτουργίας των αλγορίθμων.
5. Δύσκολα σενάρια στα οποία πρέπει να ανταποκριθούν οι αλγόριθμοι.
6. Γιατί είναι απαραίτητα όλα τα συστατικά των αλγορίθμων (γιατί χρειάζονται όλες οι γραμμές κώδικα, κλπ.) (εξήγηση με παραδείγματα ή με λόγια).
7. Διαισθητική περιγραφή της ανάλυσης των αλγορίθμων.
8. (Αν υπάρχει χώρος) 1-2 μικρές αποδείξεις.
9. Συμπεράσματα – Ανοιχτά Προβλήματα

Σε μια παρουσίαση χρησιμοποιούμε μικρές (επιγραμματικές) φράσεις και όχι μεγάλες προτάσεις. Έχουμε **πολλά σχήματα και παραδείγματα**. Σε κάθε διαφάνεια εξηγούμε πολύ αναλυτικά με λόγια τα σημεία που παρουσιάζουν ενδιαφέρον (και όχι απλά διαβάζουμε αυτά που είναι γραμμένα στη διαφάνεια).