Ενσωματωμένα Συστήματα Πραγματικού Χρόνου
Πρώτη εργασία

Φανάκης Απόστολος, 8261

apostolof@ece.auth.gr

10 Μαΐου 2019

1 Εισαγωγή

Η παρούσα εργασία εκπονήθηκε στα πλαίσια του μαθήματος "Ενσωματωμένα Συστήματα Πραγματικού Χρόνου" του ενάτου ($8^{ου}$) εξαμήνου του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Αριστοτέλειου Πανεπιστημίου Θεσσαλονίκης. Ζητούμενο του πρώτου παραδοτέου της εργασίας είναι η πειραματική υλοποίηση προγράμματος σε C με στόχο τη δειγματοληψία ανά τακτά χρονικά διαστήματα με την ελάχιστη δυνατή απόκλιση από τη πραγματική περίοδο δειγματοληψίας. Στην αναφορά αναλύονται τα πειράματα που υλοποιήθηκαν και τα αποτελέσματά τους.

2 Κώδικας

Ο κώδικας είναι διαθέσιμος, μέσω ενός gitlab repository, στη παρακάτω διεύθυνση.

https://gitlab.com/apostolof-ece-auth-gr/authRTESTask1

Η δομή των αρχείων είναι ως εξής:

- Ο φάκελος report περιέχει τον κώδικα σε latex της παρούσας αναφοράς
- Το αρχείο test_sample.c περιέχει την συνάρτηση main
- Το αρχείο test_sample_functions.h περιέχει τις δηλώσεις των συναρτήσεων, σταθερών, δομών (structs) και άλλες παραμέτρους του προγράμματος
- Το αρχείο test_sample_functions.c περιέχει τις υλοποιήσεις των συναρτήσεων

Παρέχεται επίσης αρχείο Makefile για εύκολο compilation του προγράμματος χρησιμοποιώντας την εντολή `make`.

Το πρόγραμμα δέχεται τρεις προαιρετικές παραμέτρους και εκτελείται με την εντολή `./test_sample [-t time] [-d delta] [-o output]`, όπου:

- **time** είναι ο χρόνος δειγματοληψίας σε δευτερόλεπτα για τον οποίο τρέχει κάθε πείραμα, προκαθορισμένη τιμή 7200 δευτερόλεπτα
- **delta** είναι η περίοδος δειγματοληψίας σε δευτερόλεπτα, προκαθορισμένη τιμή 0,1 δευτερόλεπτα
- **output** είναι το όνομα του αρχείου εξόδου, προκαθορισμένο όνομα αρχείου "sample_test_output"

Οι βασικές συναρτήσεις που υλοποιούν τα πειράματα και ενδιαφέρουν τον αναγνώστη είναι η συνάρτηση `main` στο αρχείο `test_sample.c` και η συνάρτηση `testSampling` στο αρχείο `test_sample_functions.c`. Περισσότερες πληροφορίες για την `testSampling` δίνονται στο επόμενο κεφάλαιο.

3 Πειράματα

Στο πρόγραμμα που παραδίδεται υλοποιούνται τέσσερα πειράματα. Όλα τα πειράματα υλοποιούνται διαδοχικά στο σώμα της συνάρτησης `testSampling`.

Στο πρώτο πείραμα (lines 29-38) γίνεται χρήση μόνο της συνάρτησης sleep με στόχο την αναστολή της εκτέλεσης του προγράμματος για χρόνο ίσο με τη περίοδο δειγματοληψίας. Έπειτα εξάγεται και αποθηκεύεται ένα timestamp. Η διαδικασία αυτή γίνεται επαναληπτικά έως ότου ξεπεραστεί ο επιθυμητός χρόνος δειγματοληψίας.

Στο δεύτερο πείραμα (lines 40-52) ακολουθείται η ίδια διαδικασία με το προηγούμενο πείραμα, αυτή τη φορά όμως διατηρείται ο μέσος όρος του σφάλματος μεταξύ της επιθυμητής και της πραγματικής περιόδου δειγματοληψίας. Ο μέσος όρος αυτός αφαιρείται από την περίοδο δειγματοληψίας κατά την κλήση της συνάρτησης sleep, με τον τρόπο αυτό επιδιώκεται η διόρθωση των αποκλίσεων της πραγματικής περιόδου δειγματοληψίας από την επιθυμητή. Μία μέθοδος που πιθανόν να επέφερε καλύτερα αποτελέσματα είναι η υλοποίηση ενός φίλτρου Wiener.

Στο τρίτο πείραμα (lines 54-82) γίνεται χρήση διακοπών για την επίτευξη του ίδιου σκοπού. Το πρόγραμμα θέτει μία επαναλαμβανόμενη διακοπή, η οποία έρχεται με τη μορφή σήματος (signal). Ορίζεται μία σημαία `sample_flag` η οποία γίνεται αληθής (true) κατά τη λήψη σήματος από διακοπή. Η διεργασία (process) ελέγχει τη σημαία επαναληπτικά. Όταν η σημαία γίνει αληθής εξάγει και αποθηκεύει το timestamp και θέτει ξανά τη σημαία ψευδή (false).

Στο τέταρτο πείραμα (lines 85-105) γίνεται χρήση τόσο της συνάρτησης sleep όσο και διακοπών. Το πείραμα θέτει και πάλι μία επαναλαμβανόμενη διακοπή. Ωστόσο, κατά την αναμονή σήματος διακοπής η διεργασία αναστέλλεται με χρήση της συνάρτησης sleep. Το πείραμα βασίζεται στο γεγονός ότι τα εισερχόμενα σήματα, αυτό της διακοπής στη προκειμένη περίπτωση, επαναφέρουν τη διεργασία του προγράμματος στο προσκήνιο, πρακτικά ακυρώνοντας τη διαδικασία sleep. Με τη τροποποίηση αυτή είναι επιθυμητή η κατανάλωση λιγότερης ενέργειας.

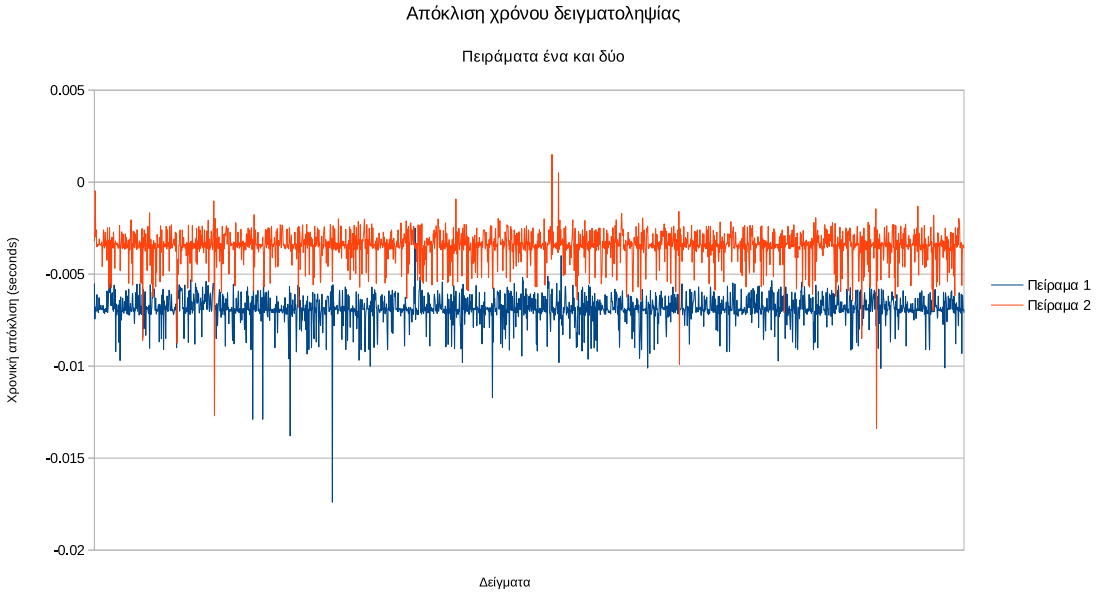
4 Αποτελέσματα

Παρακάτω φαίνονται τα αποτελέσματα των πειραμάτων για εκτέλεση δειγματοληψίας διάρκειας δύο ωρών με περίοδο 0,1 δευτερόλεπτα.

Στατιστική	Πείραμα #			
	1	2	3	4
Ελάχιστο	0.100073	0.099967	0.091687	0.092948
Μέγιστο	0.132050	0.104576	0.108309	0.106863
Μέσος όρος	0.100207	0.100101	0.09999905	0.09999974
Διάμεσος	0.100203	0.100099	0.099998	0.099999
Τυπική απόκλιση	0.000146	0.00006	0.000154	0.099989

Πίνακας 1: Στατιστικές αποτελεσμάτων πειραμάτων. Όλα τα αποτελέσματα του πίνακα έχουν μονάδα μέτρησης το δευτερόλεπτο.

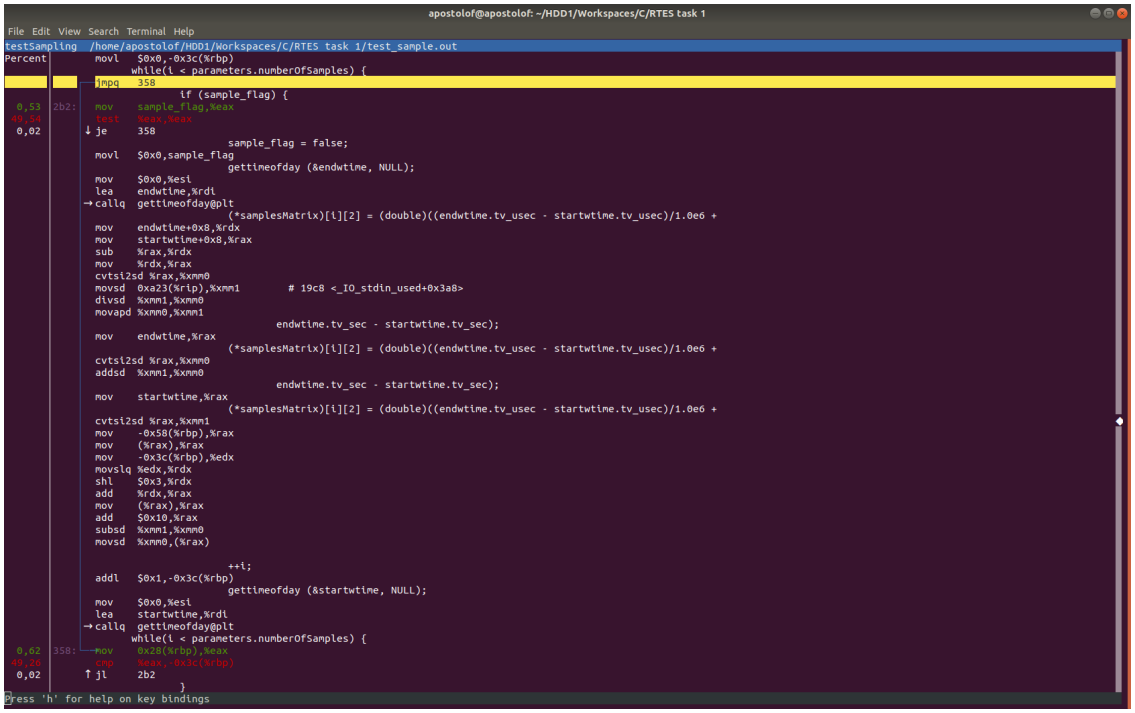
Όπως φαίνεται από τον πίνακα 1 η μέθοδος της αντιστάθμισης της περιόδου δειγματοληψίας με βάση το μέσο όρο του σφάλματος στο δεύτερο πείραμα ήταν επιτυχής, καθώς το πείραμα απέδωσε καλύτερα αποτελέσματα από το πρώτο. Ακόμα, τα πειράματα που κάνουν χρήση διακοπών (τρίτο και τέταρτο πείραμα) απέδωσαν καλύτερα από αυτά με την απλή χρήση sleep (πρώτο και δεύτερο πείραμα).



Σχήμα 1: Απόκλιση χρόνων δειγματοληψείας για τα πειράματα ένα και δύο

Σχήμα 2: Απόκλιση χρόνων δειγματοληψείας για τα πειράματα τρία και τέσσερα

Τέλος παρατηρώντας τη χρήση του επεξεργαστή κατά τη διάρκεια των πειραμάτων γίνεται εμφανές ότι η μέθοδος του τέταρτου πειράματος οδηγεί σε σαφώς μικρότερη χρήση επεξεργαστικών πόρων και άρα λιγότερη κατανάλωση ενέργειας. Το ίδιο μπορεί να παρατηρηθεί κάνοντας profiling του προγράμματος. Παρακάτω φαίνεται το ποσοστό χρήσης επεξεργαστικής ισχύος, επί της συνολικής ισχύς που χρησιμοποιήθηκε για μία εκτέλεση του προγράμματος, όπου παρατηρείται ότι η πλειοψηφία των υπολογισμών γίνεται στην επαναληπτική διαδικασί του τρίτου πειράματος.



Σχήμα 3: Profile προγράμματος