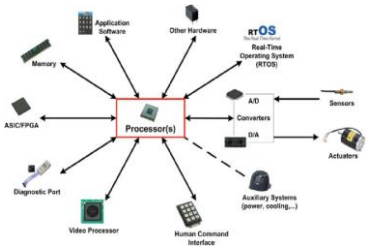


Ενσωματωμένα συστήματα πραγματικού χρόνου: Timer Producer-consumer

Φοιτητής: Μιχάλαινας Εμμανουήλ (9070)

michalain@ece.auth.gr

Καθηγητές: Πιτσιάνης Νίκος, Φλώρος Δημήτρης

Ημερομηνία: 24 Ιουλίου 2020

Κώδικας: https://github.com/manolismih/prod-cons

Περίληψη: Η παρούσα αναφορά παρουσιάζει τα σχόλια και τα συμπεράσματα που προέκυψαν από την εκπόνηση της απαλλακτικής εργασίας του μαθήματος “Ενσωματωμένα συστήματα πραγματικού χρόνου” - Τομέας Ηλεκτρονικής και Υπολογιστών – 8ο εξάμηνο. Ζητείται η υλοποίηση σε Rasberry Pi Zero ενός προγράμματος χρονιστή (timer), η λειτουργία του οποίου βασίζεται σε pthreads. Πιο συγκεκριμένα, κάθε χρονιστής δημιουργεί ένα νήμα που εκτελεί την συνάρτηση παραγωγού (producer). Η συνάρτηση εισάγει περιοδικά συναρτήσεις προς εκτέλεση σε μια κοινή ουρά FIFO. Ταυτόχρονα, άλλα νήματα εκτελούν την συνάρτηση καταναλωτή (consumer), η οποία εξάγει και εκτελεί τις αποθηκευμένες συναρτήσεις στην ουρά. Στο τέλος, πραγματοποιούνται πειράματα με μεταβλητό αριθμό χρονιστών, περιόδου και φόρτου ανα κλήση της συνάρτησης προς εκτέλεσης και παρουσιάζονται στατιστικά, όπως η διολίσθηση (drift) μεταξύ των διαδοχικών εκτελέσεων, ο χρόνος που ξοδεύουν οι συναρτήσεις παραγωγού και καταναλωτή και η χρήση CPU της πλακέτας.

Λέξεις-κλειδιά: pthread, producer-consumer, FIFO ουρά, drift, timer, Rasberry Pi Zero.

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

1. ΔΟΜΗ ΤΟΥ PROJECT – CODE REPO

prod-cons.c: Το κεντρικό πηγαίο αρχείο, περιέχει την main, την συνάρτηση καταναλωτή και τον κώδικα για αρχικοποίηση και εκκίνηση των χρονιστών.

queue.c: Περιέχει κώδικα που αποτελεί την λειτουργικότητα της FIFO ουράς.

functions.c: Περιέχει συναρτήσεις που εισάγονται στην FIFO ουρά, καθώς και συναρτήσεις που παράγουν τυχαία ορίσματα και βοηθητικά struct.

timer.c: Περιέχει κώδικα που αποτελεί την λειτουργικότητα των χρονιστών και την συνάρτηση παραγωγού.

pc.c: Το αρχικό παράδειγμα, για λόγους πληρότητας.

Δεδομένα εξόδου

ProducerTimeLog.txt: Καταγράφει τον χρόνο που χρειάστηκε κάθε συνάρτηση εκτέλεσης για να εισαχθεί στην ουρά από τον παραγωγό.

ConsumerTimeLog.txt: Καταγράφει τον χρόνο που χρειάστηκε ο καταναλωτής για να **αφαιρέσει** και να εκτελέσει **εκάστη** συνάρτηση από την ουρά.

Drift###msLog.txt: Καταγράφει την καθυστέρηση έναρξης εκτέλεσης μιας συνάρτησης σε σχέση με την προηγούμενη και την θεωρητική περίοδο με την οποία πρέπει να εκτελείται. Θετικές τιμές σημαίνουν ότι υπήρξε μια καθυστέρηση, ενώ αρνητικές σημαίνουν ότι οι συγκεκριμένες εκτελέσεις εκ**κ**ινήθηκαν νωρίτερα από την προγραμματισμένη τους περίοδο.

2. ΠΕΙΡΑΜΑΤΑ ΧΡΟΝΙΣΤΩΝ ΜΕ ΠΕΡΙΟΔΟΥΣ 10, 100 ΚΑΙ 1000 MS

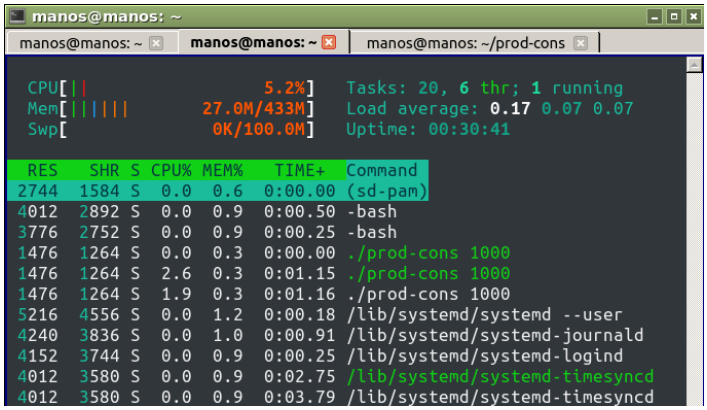
Σε αυτά τα πειράματα χρησιμοποιείται ως συνάρτηση προς εκτέλεση η mult_100x100_matrix, η οποία πολλαπλασιάζει δύο πίνακες 100 επί 100. Το Rasberry Pi Zero έχει μόνο έναν πυρήνα με 1 thread, και επειδή οι παραγωγοί αναλώνουν την περισσότερη ώρα εκτέλεσης τους με την εντολή usleep και δεν καταναλώνουν χρόνο CPU, σχεδόν όλος ο χρόνος CPU καταναλώνεται από τους καταναλωτές. Αρα, για βέλτιστη απόδοση αρκεί να υπάρχει μόνο ένας καταναλωτής, καθώς περισσότεροι απλά θα ανταγωνίζονται μεταξύ τους στην χρήση CPU και θα δημιουργούν επιπλέον καθυστέρηση (overhead) λόγω της ανάγκης του λειτουργικού να μοιράζει τον χρόνο CPU σε επιπλέον νήματα.

Σημείωση: Σε όλους τους πίνακες που ακολουθούν, οι χρόνοι είναι εκφρασμένοι σε μικροδευτερόλεπτα (μs).

Χρονιστής με περίοδο 1000ms

Σε αυτό το πείραμα η ουρά μένει συνεχώς άδεια, οπότε δεν προκαλείται σημαντική καθυστέρηση στην εκτέλεση των συναρτήσεων. Η χρήση CPU κυμαίνεται γύρω στο 5%.

	Ελάχιστο	Μέγιστο	Μέσος όρος	Διάμεσος	Τυπική απόκλιση
Καθυστέρηση παραγωγού	29	154	39.4	39	7.1
Χρόνος καταναλωτή	11179	36451	16391.8	15966	2455.9
Διολίσθηση	47	9661	187.8	165	369.8



Σχήμα 1: Επισκόπηση της εκτέλεσης του προγράμματος με το htop. Φαίνονται 2 νήματα prod-cons (πράσινα γράμματα), ένα για τον χρονιστή και ένα για τον καταναλωτή. Η διεργασία με τα άσπρα γράμματα είναι η πατρική.

Χρονιστής με περίοδο 100ms

Σε αυτό το πείραμα επίσης η ουρά μένει συνεχώς άδεια, αλλά η χρήση της CPU κυμαίνεται γύρω στο 17%.

	Ελάχιστο	Μέγιστο	Μέσος όρος	Διάμεσος	Τυπική απόκλιση
Καθυστέρηση παραγωγού	22	269	32.4	34	7
Χρόνος καταναλωτή	11007	43288	14678	15817	3171.9
Διολίσθηση	-125	11693	165.9	158	165.9

Χρονιστής με περίοδο 10ms

Σε αυτό το πείραμα η χωρητικότητα της ουράς (1000 στοιχεία) γεμίζει γρήγορα. Έτσι ο παραγωγός καθυστερεί σημαντικά περισσότερο για να εισάγει συναρτήσεις σε αυτήν, αφού πρέπει να περιμένει να αδειάσει κάποια θέση. Οι τιμές διολίσθησης είναι επίσης αυξημένες, διότι η χρήση της CPU είναι σταθερά στο 100% την οποία σχεδόν μονοπωλεί ο καταναλωτής. Παρόλα αυτά, η πλακέτα δεν έχει αρκετή υπολογιστική ισχύ για να εκτελεί τις εργασίες έγκαιρα.

	Ελάχιστο	Μέγιστο	Μέσος όρος	Διάμεσος	Τυπική απόκλιση
Καθυστέρηση παραγωγού	3	78475	11656.9	12908	4514.2
Χρόνος καταναλωτή	10943	78467	12940.8	12876	1791.4
Διολίσθηση	968	68500	2982.9	2917	1806.7

Χρονιστές με περιόδους 1000, 100 και 10 ms που τρέχουν ταυτόχρονα

Σε αυτό το πείραμα η χρήση της CPU είναι επίσης στο 100%. Η υπολογιστική ισχύς της πλακέτας δεν επαρκεί για την έγκαιρη έναρξη των συναρτήσεων, κάτι που φαίνεται από τους πολύ υψηλούς μέσους όρους των χρόνων διολίσθησής. Η καθυστέρηση του παραγωγού είναι επίσης μεγάλη επειδή η ουρά γεμίζει γρήγορα.

Για τον χρονιστή των 1000ms παρατηρούμε διολισθήσεις έως και 485ms. Στους χρονιστές 100ms και 10ms βλέπουμε μέγιστες διολισθήσεις 89ms και 69ms αντίστοιχα καθώς και μέσο όρο διολίσθησης χρονιστή 10ms τα 4.5ms. Η εκτέλεση των συναρτήσεων συνεπώς δεν είναι καθόλου περιοδική.

Αξιοσημείωτες είναι οι πολύ μικρές ελάχιστες (αρνητικές) τιμές στους χρόνους διολίσθησης. Καθώς η εκτέλεση των συναρτήσεων δεν είναι πια περιοδική, μια εκτέλεση

που θα καθυστερήσει υπερβολικά να ξεκινήσει θα οδηγήσει σε μια επόμενη εκτέλεση σε χρόνο μικρότερο από την περίοδο, διότι ο προγραμματισμός γίνεται με βάση μία σταθερή καθυστέρηση. Η μη περιοδικότητα αποτυπώνεται στις μεγάλες τυπικές αποκλίσεις των διολισθήσεων, που είναι πολλαπλάσιες από τις αντίστοιχες στις μεμονωμένες εκτελέσεις.

	Ελάχιστο	**Μέγιστο**	**Μέσος όρος**	**Διάμεσος**	**Τυπική απόκλιση**
Καθυστέρηση παραγωγού	2	77893	14574.4	12966	5830.5
Χρόνος καταναλωτή	10930	58290	13095.3	12877	1989.5
Διολίσθηση χρονιστή 1000ms	-91850	485014	15041.5	1556	70224.2
Διολίσθηση χρονιστή 100ms	-60433	88760	1781.5	3944	13421.4
Διολίσθηση χρονιστή 10ms	953	68997	4582.3	2921	4885.4

3. ΠΕΙΡΑΜΑ ΛΕΙΤΟΥΡΓΙΑΣ ΠΡΑΓΜΑΤΙΚΟΥ ΧΡΟΝΟΥ

Και πάλι θα έχουμε **ένα μόνο νήμα καταναλωτή**, ενώ επιπλέον θέλουμε η εκτέλεση των εργασιών να γίνεται έγκαιρα, άρα θα πρέπει **να μην γεμίζει ποτέ η ουρά.** Αν ο ρυθμός παραγωγής είναι μεγαλύτερος από τον ρυθμό κατανάλωσης, τότε η ουρά στην μακροχρόνια λειτουργία, ασχέτως του μεγέθους της θα γεμίσει, οδηγώντας σε καθυστερήσεις. Επομένως, οι μοναδικές παράμετροι που καθορίζουν την ευστάθεια του συστήματος είναι (υποθέτοντας ότι εκτελείται μόνο ένας χρονιστής) η περίοδος του χρονιστή και η διάρκεια εκτέλεσης της συνάρτησης. Με δεδομένη την συνάρτηση εκτέλεσης mult_100x100_matrix, ψάχνουμε να βρούμε την **ελάχιστη περίοδο για την οποία η ουρά διατηρείται άδεια.**

Από τα προηγούμενα πειράματα γνωρίζουμε ότι για περίοδο 100ms η ουρά είναι συνεχώς άδεια, ενώ για περίοδο 10ms η ουρά γεμίζει. Με δοκιμές βρίσκουμε ότι για περίοδο 13ms το αποτέλεσμα είναι αμφίρροπο, ενώ **για περιόδους 14ms και πάνω η ουρά είναι συνεχώς άδεια.**