

Μεταπτυχιακό Μάθημα ΟΣΥΛ
“Λειτουργικά Συστήματα Πραγματικού Χρόνου”
Μ.Στεφανιδάκης

Εργαστηριακή Άσκηση #2

Σκοπός της παρούσας άσκησης είναι η εξοικείωση με τις διάφορες μεθόδους χρονοδρομολόγησης στατικών και δυναμικών προτεραιοτήτων.

Εισαγωγή: Η εφαρμογή σας θα χρησιμοποιήσει τη βιβλιοθήκη LXRT του interface πραγματικού χρόνου RTAI στο λειτουργικό σύστημα Linux. Η βιβλιοθήκη αυτή επιτρέπει την ανάπτυξη εφαρμογών πραγματικού χρόνου σε **user space**. Για την δημιουργία της εφαρμογής σας θα συνδυάσετε τα εξής αρχεία:

α) βιβλιοθήκες RTAI: **rtai_hal rtai_lxrt rtai_sem rtai_msg rtai_mbx** (τα modules αυτά θα πρέπει να υπάρχουν στη μνήμη του συστήματος).

β) κώδικας εφαρμογής: **rt_process.c rt_process.h task.c display.c display.h** (σε 3 διαφορετικά directories: **simple, fixprio** και **edf**).

Ο κώδικας που θα τροποποιήσετε βρίσκεται στα αρχεία **rt_process.c** και **task.c**, όπου θα βρείτε λεπτομερείς οδηγίες για τα ερωτήματα που ακολουθούν.

1. Συνδεθείτε στο σύστημα του εργαστηρίου όπου έχει εγκατασταθεί το interface RTAI ως χρήστης **rtaiuser**. Δημιουργήστε ένα directory στο ίδιο επίπεδο με τα **simple, fixprio** και **edf**.

2. Αντιγράψτε τα αρχεία από το directory **simple** στο δικό σας. Η εφαρμογή δημιουργεί μία **περιοδική** διεργασία με ($T_1=2ms, C_1\approx 1ms$). Το αρχείο **rt_process.c** περιέχει τη συνάρτηση **main** του εκτελούμενου κώδικα, η οποία δημιουργεί τη διεργασία πραγματικού χρόνου ως **παράλληλο thread**. Ο κώδικας του thread βρίσκεται στο αρχείο **task.c**.

α) Μελετήστε τη δομή των δύο αρχείων.

β) Μεταγλωττίστε και εκτελέστε το πρόγραμμα δοκιμαστικά. Τι παρατηρείτε ως προς την εκκίνηση της διεργασίας και τον χρόνο εκτέλεσης;

3. Αντιγράψτε τα αρχεία από το directory **fixprio** στο δικό σας. Η εφαρμογή δημιουργεί δύο περιοδικές διεργασίες με τα εξής χαρακτηριστικά: ($T_1=2ms, C_1\approx 1ms$) ($T_2=10ms, C_2\approx 4.5ms$). Τα χαρακτηριστικά αυτά ορίζονται στο αρχείο **rt_process.c** σε δομή που περιγράφει **<id:όνομα:προτεραιότητα:φ(ns):T(ns):C(ns):αριθμός-περιόδων>** ανά διεργασία, όπου **id = 0...n-1** (χρειάζεται μόνο για το σύστημα καταγραφής) και **όνομα = string** έως 6 χαρακτήρες, πρέπει να είναι μοναδικό (απαίτηση RTAI).

α) Ποιο είναι το μέγεθος της χρησιμοποίησης U;

β) Μπορεί ο αλγόριθμος RM να εγγυηθεί εφικτό πρόγραμμα χρονοδρομολόγησης;

γ) Ποια διεργασία έχει τη μεγαλύτερη προτεραιότητα σύμφωνα με τον RM;

δ) Εκτελέστε την εφαρμογή με την κατάλληλη προτεραιότητα. Τι παρατηρείτε;

ε) Αναστρέψτε τις προτεραιότητες και εκτελέστε πάλι την εφαρμογή.

4. Αντιγράψτε το αρχείο **rt_process.c** από το directory **edf** στο δικό σας. Η εφαρμογή δημιουργεί τώρα δύο περιοδικές διεργασίες με τα εξής χαρακτηριστικά: ($T_1=10ms, C_1\approx 5ms$) ($T_2=25ms, C_2\approx 12ms$).

α) Ποια η χρησιμοποίηση U;

β) Δοκιμάστε να αναθέσετε προτεραιότητες και να εκτελέσετε τις διεργασίες σύμφωνα με τον αλγόριθμο RM. Τι παρατηρείτε;

γ) Αντιγράψτε το αρχείο **task.c** από το directory **edf** στο δικό σας, έτσι ώστε να επιτύχετε χρονοδρομολόγηση EDF (συνάρτηση **rt_task_set_resume_end_times**). Εκτελέστε την εφαρμογή. Επιτυγχάνεται ορθή χρονοδρομολόγηση τώρα;