

Χρονοδρομολόγηση II

Χρονοδρομολόγηση off-line – Σποραδικές διεργασίες

Μ.Στεφανιδάκης

Μέθοδοι χρονοδρομολόγησης (επανάληψη)

- Χρονοδρομολόγηση on-line
 - επιλογή κατά τη διάρκεια της εκτέλεσης
 - στατικές και δυναμικές προτεραιότητες
- Υπολογισμός off-line
 - κατασκευή πίνακα χρονικής εκτέλεσης
 - αποφάσεις χρονοδρομολόγησης σε προκαθορισμένα χρονικά διαστήματα (clock-driven)
 - χρήση πολύπλοκων αλγορίθμων
 - γνωστά χαρακτηριστικά διεργασιών

Χρονοδρομολόγηση off-line

- προϋποθέσεις
 - σταθερός αριθμός περιοδικών διεργασιών
 - γνωστές παράμετροι εκτέλεσης
 - περίοδος (T_i), χρόνος εκτέλεσης (C_i), προθεσμία (D_i)
 - αμελητέα απόκλιση χρόνου έκδοσης
 - πάντα ανά περίοδο
- περιοδικό πρόγραμμα εκτέλεσης
 - κυκλικές διεργασίες
 - πολλαπλοί ρυθμοί
- παραδείγματα κυκλικών διεργασιών
 - ανάκτηση πληροφορίας από αισθητήρες
 - βρόχος έλεγχου συστημάτων

Στατικό πρόγραμμα εκτέλεσης

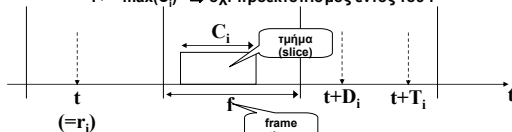
- Διεργασίες με διαφορετικούς ρυθμούς (περιόδους)
 - Υπερπερίοδος (H): ελάχιστο κοινό πολλαπλάσιο περιόδων
 - Μέγιστο όριο αριθμού διεργασιών εντός του H:

$$N = \sum_i \frac{H}{T_i}$$

- Πρόγραμμα εκτέλεσης: περιγράφει μια υπερπερίοδο
- Οργάνωση σε F ίσα χρονικά διαστήματα (frames) ανά H
 - F ακέραιο υποπολλαπλάσιο H
 - αποφάσεις χρονοδρομολόγησης: στην έναρξη κάθε frame

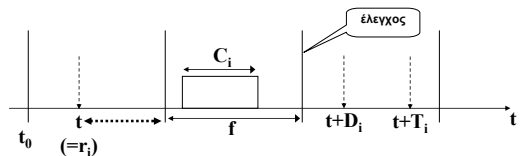
Στατικό πρόγραμμα εκτέλεσης

- Σχεδιασμός
 1. επιλογή μεγέθους frame
 2. κατανομή έργου διεργασιών σε τμήματα (slices)
 3. τοποθέτηση τμημάτων σε frames
- Εύρεση μεγέθους frame f:
 1. f ακέραιο υποπολλαπλάσιο H
 2. κάθε τμήμα πρέπει να χωράει σε ένα frame
 - $f \geq \max(C_i) \Rightarrow$ όχι προεκτοπισμός εντός του f



Στατικό πρόγραμμα εκτέλεσης

- Εύρεση μεγέθους frame f (συνέχεια):
 3. πρέπει να υπάρχει δυνατότητα ελέγχου της εκτέλεσης
 - δηλ. εάν η εκτέλεση κάθε τμήματος έγινε εντός προθεσμίας
 - αποφάσεις και έλεγχος μόνο στην έναρξη κάθε νέου frame
 - Ένα τουλάχιστον frame ανάμεσα σε t και t + D_i
 - Ισχύει όταν $D_i \geq 2f - \mu.κ.δ.(T_i, f)$
 - όταν $D_i \geq f$ εάν $t_0 = t$



Στατικό πρόγραμμα εκτέλεσης

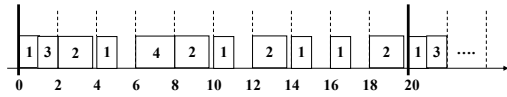
($T_1=4$ $C_1=1$), ($T_2=5$ $C_2=1.8$), ($T_3=20$ $C_3=1$), ($T_4=20$ $C_4=2$)

• $\max(C_i) = 2 : f \geq 2$

• $H = 20 : f = 2, 4, 5, 10, 20$

• Για κάθε i πρέπει $D_i \geq 2f - \mu.κ.δ.(T_i, f) : f = 2$

παράδειγμα κατανομής



- Εύρεση κατανομής όχι πάντοτε εφικτή
 - υποδιαίρεση tasks σε μικρότερα μέρη;
 - αλλά: περισσότερος χρόνος για context switching
 - αλγοριθμική εύρεση: network-flow graphs

Αλγοριθμική εύρεση προγράμματος εκτέλεσης

- Επιλογή frame-time και κατανομή τμημάτων διεργασιών

- Στη γενική μορφή NP-hard πρόβλημα
- Iterative Network-flow (INF) αλγόριθμος
 - Εξέταση όλων των πιθανών f
 - Network-flow graph, κορυφές jobs και frames
 - Ροή ακμών, χρόνος του job _{x} στο frame _{x}
 - Εύρεση μέγιστης ροής
 - $O((N+F)^3)$ για N jobs και F frames

Κυκλική εκτέλεση

- Υποθέσεις:

- ύπαρξη timer interrupt
- ελάχιστη καθυστέρηση (με γνωστό άνω όριο)

- Πρόγραμμα εκτέλεσης

- Πίνακας Π , F γραμμών
- εκτέλεση ανά frame

- Έλεγχος εκτέλεσης

- frame overruns

- Μη περιοδικές διεργασίες;

```

cyclic_scheduler() {
    for ever {
        wait_on_timer_interrupt();
        εάν εκτελείται ακόμα task
        τότε σφάλμα
        F=F+1 // + wrap-around
        εάν δεν έχουν εκδοθεί όλα
        τα tasks του  $\Pi(F)$ 
        τότε σφάλμα
        Εκτέλεση όλων των tasks του
         $\Pi(F)$  // ή ξεχωριστός server
    }
}
    
```

Μη περιοδικές διεργασίες

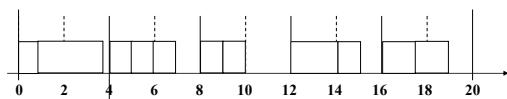
- A. Μη-περιοδικά tasks χωρίς προθεσμία

- άγνωστος χρόνος έκδοσης (άφιξης)
- event-driven (interrupts)
- στόχος: μείωση χρόνου απόκρισης (response time)

- μέθοδοι εξυπηρέτησης

- background execution
 - όταν υπάρχουν κενά στο frame
- slack stealing
 - βελτίωση χρόνου απόκρισης

Εκτέλεση μη περιοδικών διεργασιών



background execution

slack stealing

- Slack για κάθε frame = γνωστό
 - Στην αρχή frame και σε κάθε τεμαχισμό slice:
 - Εκτέλεση μη περιοδικών διεργασιών μέχρι εξάντλησης του slack
 - Ακρίβεια μέτρησης χρόνου; (timer)

Μη περιοδικές διεργασίες

- B. Μη-περιοδικά tasks με προθεσμίες (sporadic)

- άγνωστος χρόνος έκδοσης (άφιξης)
- αλλά: γνωστός χρόνος εκτέλεσης (C_i)

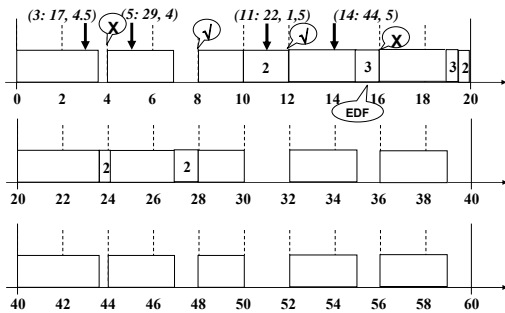
1. Acceptance test ($O(N_s)$)

- σε σχέση με περιοδικά tasks και άλλα ήδη υπάρχοντα μη-περιοδικά με προθεσμία
- διαθέσιμος χρόνος + προθεσμίες μη-περιοδικών tasks

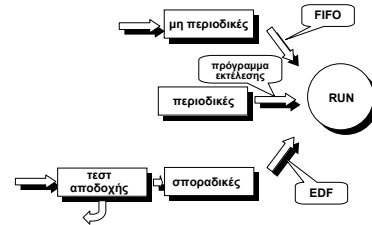
2. Χρονοδρομολόγηση

- τύπου EDF
- ουρά μη-περιοδικών tasks με προθεσμίες

Εκτέλεση μη περιοδικών διεργασιών



Συστήματα με μικτού τύπου διεργασίες



Αλλαγή λειτουργίας

- **Mode change**
- **Αναπροσαρμογή του συστήματος**
 - αλλαγή αριθμού διεργασιών
 - διαγραφή παλιών - προσθήκη νέων διεργασιών
 - νέο πρόγραμμα εκτέλεσης
 - υπολογισμός off-line
 - διεργασία αλλαγής λειτουργίας
 - υψηλή προτεραιότητα
 - με ή χωρίς προθεσμία
 - καθυστέρηση μη-περιοδικών διεργασιών

Αντιμετώπιση σφαλμάτων

- **Frame overruns**
 - υπέρβαση χρόνου frame
 - "σφάλμα" υλικού ή λογισμικού
 - από συνδυασμό δεδομένων εισόδου που δεν έχουν προβλεφθεί
 - από (μεταβατικό) σφάλμα υλικού που καθυστερεί την εκτέλεση
 - από σφάλμα λογισμικού που δεν έχει ανιχνευθεί κατά την ανάπτυξη
- **Αντιμετώπιση**
 - απόρριψη διεργασίας
 - καθυστέρηση = παραγωγή εσφαλμένων αποτελεσμάτων
 - ολοκλήρωση εκτέλεσης
 - ενδεχόμενη καθυστέρηση άλλων διεργασιών
 - απαραίτητα αποτελέσματα
 - συνέχιση ως μη-περιοδική χωρίς προθεσμία
 - σε χρονικά κενά εκτέλεσης

Πλεονεκτήματα και Μειονεκτήματα στατικής εκτέλεσης

- **Πλεονεκτήματα**
 - Απλότητα, προκαθορισμένη λειτουργία
 - Πλήρης υπολογισμός όλων των περιορισμών off-line
 - Λεπτομερής συγχρονισμός διεργασιών
 - Επιβαρύνσεις μικρές
 - Ευκολός έλεγχος ορθότητας
- **Μειονεκτήματα**
 - Απαιτείται πλήρης γνώση χαρακτηριστικών διεργασιών
 - και: όλων των πιθανών αλληλεπιδράσεων μεταξύ τους
 - Δυσκολία τροποποίησης (mode change)

Χρήσεις time-driven μοντέλου

- Σε συστήματα αποτελούμενα από πολλά τμήματα
 - Διαφορετικοί κατασκευαστές = κάθε τμήμα ελέγχεται ατομικά ως προς τη λειτουργία του σε πραγματικό χρόνο
 - Time-driven (*time-triggered*) λειτουργία και επικοινωνία: διασφάλιση ορθής συνολικής λειτουργίας σε πραγματικό χρόνο
 - επικοινωνία σε καθορισμένα χρονικά όρια
 - αποφυγή χρονικών ασαφειών
 - μείωση σφαλμάτων
 - εγγυημένος συγχρονισμός
 - χωρίς διαίτησια για χρήση κοινών πόρων