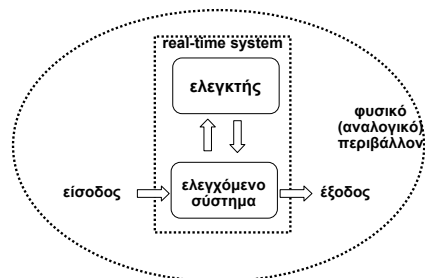


Λειτουργικά Συστήματα Πραγματικού Χρόνου

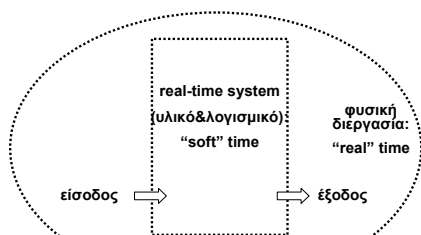
Η χρήση του χρόνου

Μ.Στεφανιδάκης

Συστήματα πραγματικού χρόνου: ελεγκτής και ελεγχόμενο σύστημα



Δύο διαφορετικά συστήματα χρόνου

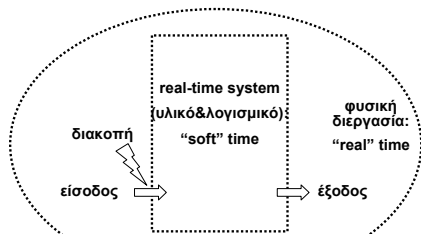


- Λειτουργικό Σύστημα Πραγματικού χρόνου:
 - Συμβιβάζει τα δύο διαφορετικά χρονικά συστήματα!

Φυσικές και υπολογιστικές διεργασίες

- Στο φυσικό περιβάλλον
 - Οι διάφορες διεργασίες εκτελούνται ταυτόχρονα
- Στο υπολογιστικό περιβάλλον
 - Υπάρχει περιορισμός πόρων εκτέλεσης
 - Περιορισμένος αριθμός CPU
 - Συχνά μόνο 1 CPU
 - Ψευδο-παράλληλη εκτέλεση διεργασιών
 - "Τεχνητός" καταμερισμός και αλληλουχία διεργασιών
 - Απόκλιση από τον "πραγματικό χρόνο"
 - Χρονικές επιβαρύνσεις για διαχείριση διεργασιών

Εξωτερικές (ασύγχρονες) διακοπές



- Εξωτερικές διακοπές (interrupts)
 - Το σημείο εισόδου του πραγματικού χρόνου στο σύστημα

Ο μηχανισμός των διακοπών

- Δημιουργία από εξωτερική συσκευή
 - Αισθητήρες και άλλα ενεργά περιφερειακά
 - Αίτηση εξυπηρέτησης!
- Συγκέντρωση σε ελεγκτή διακοπών (PIC)
 - Και τροφοδότηση σε είσοδο διακοπών επεξεργαστή
 - Maskable/Non-maskable interrupts
- Ο επεξεργαστής διακόπτει την τρέχουσα εκτέλεση
 - Το ακριβές σημείο διακοπής καθορίζεται από την αρχιτεκτονική
 - Αποθήκευση τρέχουσας κατάστασης εκτέλεσης

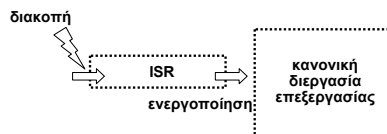
Ο μηχανισμός των διακοπών

- **Ρουτίνα Εξυπηρέτησης Διακοπής (ISR)**
 - Εκτέλεση μέσω ενός πίνακα διανυσμάτων διακοπών
 - Όσο πιο σύντομη γίνεται
 - Χρήση τρέχοντος ή ειδικού χώρου stack
 - Τι συμβαίνει αν επιτρέπονται nested interrupts;
 - Είναι re-entrant;
- **Επιστροφή στη διακοπεΐσα διεργασία**
 - Μετά το πέρας της εκτέλεσης της ISR

Απόκριση ΛΣΠΧ σε διακοπές

- Πρέπει να είναι μετρήσιμη (ή φραγμένη από άνω όριο)
 - Αναγκαίο για εφαρμογές πραγματικού χρόνου
- **Αναγνώριση και εξυπηρέτηση διακοπών**
 - **Latency (CPU):** μέγιστος χρόνος με απενεργοποιημένα interrupts (ΛΣΠΧ!) + χρόνος εκκίνησης ρουτίνας ISR (CPU)
 - Αν εκτελείται ISR διακοπής υψηλότερης προτεραιότητας;
 - Δεν είναι φραγμένη ποσότητα!
 - **Response (ΛΣΠΧ):** latency + χρόνος αλλαγής context + χρόνος έναρξης εκτέλεσης επιθυμητής λειτουργίας
 - Δεν είναι φραγμένη ποσότητα επίσης!

Εξυπηρέτηση διακοπών



- **Η σύγχρονη τακτική σχεδίασης**
 - ISR όσο πιο σύντομη γίνεται
 - Μειώνεται η πιθανότητα να χαθούν επόμενες διακοπές
 - Επιτρέπει την εκτέλεση κώδικα μεγάλου μήκους σε χαμηλότερες προτεραιότητες

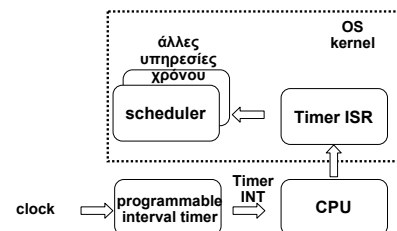
Διακοπές και προτεραιότητες

- **ISR**
 - Υψηλότερη προτεραιότητα από τις λοιπές διεργασίες
 - Ο έλεγχος και η επίτρεψη υλοποιείται μέσω υλικού
 - Ενώ για τις κοινές διεργασίες του συστήματος, μέσω λογισμικού (ΛΣ)
 - **Εξαίρεση: maskable interrupts**
 - Απενεργοποίηση μέσω λογισμικού

Η Χρήση του Χρόνου

- **Clock Tick: διακοπή που συμβαίνει περιοδικά**
 - ο “παλμός” του ΛΣΠΧ (μs - ms)
 - συχνότερα = μεγαλύτερη επιβάρυνση
- **Εκτέλεση λειτουργιών σε καθορισμένο χρονικό διάστημα**
 - Χρονοδρομολόγηση
 - Timeouts - Delays
 - Παράδειγμα χρήσης: sleep(10msec);
- **Επιτυγχάνεται ακριβής τήρηση των χρονικών διαστημάτων;**

Δημιουργία “υπολογιστικού” χρόνου



Καταμερισμός λειτουργιών εφαρμογής

- Η αντιστοίχιση λειτουργιών και διεργασιών
- Υπόψη:
 - κάθε νέο task κοστίζει: TCB space + runtime stack space
 - αλλά διευκολύνει την οργάνωση και την εκ των υστέρων κατανόηση του κώδικα
- Επιλογή λειτουργιών και διεργασιών
 - Χαρακτηριστικά λειτουργιών
 - Μεγιστοποίηση χρήσης CPU

Μεγιστοποίηση χρήσης CPU

- Πρέπει να διατηρούμε τον επεξεργαστή όσο το δυνατόν απασχολημένο
 - ή τους επεξεργαστές (συνεπάγεται καταναμημένο ΛΣΠΧ!)
- Ιδίως όταν υπάρχουν διεργασίες I/O με μεγάλους χρόνους αναμονής
- Διάσπαση κώδικα για επίτευξη επικάλυψης της αναμονής
- Συνήθως τα δεδομένα είναι διαθέσιμα σε buffers
- Εξέταση δυνατότητας εκτέλεσης λειτουργιών ανεξάρτητα από τις προηγούμενες

Αρχικός καταμερισμός

- Software Requirements Specification
 - Αποκρυσταλλώνει τις απαιτήσεις της εφαρμογής
- Αρχική κατανομή
 - "Outside-in approach"
 - Αναγνώριση εισόδων και εξόδων στο σύστημα πραγματικού χρόνου
 - Σημεία εισόδου-εξόδου δεδομένων από περιβάλλον
 - Περιγραφή ροής πληροφορίας
 - Δημιουργία διεργασιών που χειρίζονται τα σημεία εισόδου-εξόδου
 - Στη συνέχεια η ανάλυση προχωρά στη δημιουργία και σύνθεση διεργασιών υψηλότερου επιπέδου

Χαρακτηριστικά λειτουργιών

- Γενικά
 - Δεν πρέπει να μπαίνουν στην ίδια διεργασία λειτουργίες με διαφορετικά χαρακτηριστικά:
 - Τρόπο ενεργοποίησης
 - Επίπεδο προτεραιότητας
 - Χρονική κλίμακα
- Δημιουργία διεργασιών ανάλογα με
 - Λειτουργικότητα
 - Χρονικές εξαρτήσεις
 - Εξαρτήσεις από γεγονότα (και περιφερειακές συσκευές)

Λειτουργικότητα

- Δημιουργία διεργασιών ανάλογα με λειτουργικότητα
 - Κάθε κύρια λειτουργία καλό είναι να εκτελείται από ξεχωριστή διεργασία
 - Λειτουργίες στενά συνδεδεμένες μεταξύ τους, μπορούν να ανήκουν στην ίδια διεργασία.
 - Λειτουργίες που υποχρεωτικά διαδέχονται η μία την άλλη, μπορούν να μπουν στην ίδια διεργασία.
 - Μακρά επεξεργασία (π.χ. number crunching) τοποθετείται σε ξεχωριστή διεργασία χαμηλής προτεραιότητας

Χρονικές εξαρτήσεις

- Δημιουργία διεργασιών ανάλογα με χρονικές εξαρτήσεις
 - Λειτουργίες με διαφορετικές προθεσμίες κωδικοποιούνται σε διαφορετικές διεργασίες
 - Λειτουργίες κρίσιμες για το σύστημα τοποθετούνται σε υψηλής προτεραιότητας διεργασίες
 - Λειτουργίες με διαφορετική περίοδο επαναληπτικής εκτέλεσης μπορούν να ομαδοποιηθούν σε διεργασίες ανάλογα με την ομοιότητα των περιόδων τους
 - Μη συνδεδεμένες μεταξύ τους λειτουργίες που εκτελούνται όμως πάντα μαζί μπορούν να μπουν στην ίδια διεργασία

Εξαρτήσεις από γεγονότα

- Οι διεργασίες θα πρέπει να μπορούν να χειριστούν τα (ασύγχρονα) γεγονότα εντός προθεσμιών
 - Συσκευές I/O με διαφορετικούς ρυθμούς εξυπηρετούνται από διεργασίες αντίστοιχης προτεραιότητας
 - Συσκευές I/O με υψηλό ρυθμό δημιουργίας διακοπών θα πρέπει να εξυπηρετούνται από ξεχωριστή διεργασία
 - Συσκευές I/O με χαμηλό ρυθμό δημιουργίας διακοπών και μη πειστικές προθεσμίες μπορούν να εξυπηρετούνται από την ίδια διεργασία

Ενεργοποίηση διεργασιών

- **Αυτο-ενεργοποίηση σε χρονικό διάστημα**
 - Λειτουργίες ανίχνευσης εισόδων
 - Μικρή διάρκεια εκτέλεσης
 - Μεγάλη προτεραιότητα
 - Αναγνώριση και αποθήκευση των αλλαγών σε συσκευές εισόδου
 - Η επεξεργασία γίνεται αργότερα από χαμηλότερης προτεραιότητας διεργασίες

```
taskA( ) {  
  do { // for ever  
    getSensorInput();  
    sleep(10msec);  
  }  
}
```

Ενεργοποίηση διεργασιών

- **Ενεργοποίηση από άλλες διεργασίες**
 - Εκκίνηση από την αρχή μιας νέας διεργασίας
 - Η νέα διεργασία πρέπει να τερματίζει (όχι loop)
 - Πολλά ΛΣΠΧ επιτρέπουν στη μητρική διεργασία να περιμένει τον τερματισμό της θυγατρικής.
 - Συχνά μία αυτοενεργοποιούμενη διεργασία, όταν υπάρχουν νέα δεδομένα, δημιουργεί άλλη διεργασία για να τα επεξεργαστεί

```
taskA( ) {  
  do { // for ever  
    getSensorInput();  
    if (dataready)  
      taskStart(B);  
    sleep(10msec);  
  }  
}
```

```
taskB( ) {  
  <processData>  
  ...  
  selfTerminate();  
}
```

Ενεργοποίηση διεργασιών

- **Ενεργοποίηση από άλλες διεργασίες**
 - Συνέχιση μιας νέας διεργασίας
 - Η νέα διεργασία έχει ατέρμονα βρόχο
 - Η νέα διεργασία έχει ξεκινήσει νωρίτερα
 - η ενεργοποίηση της νέας διεργασίας γίνεται γρηγορότερα
 - δεν δημιουργείται εκείνη τη στιγμή, αλλά υπάρχει σε κατάσταση αναμονής

```
taskA( ) {  
  do { // for ever  
    getSensorInput();  
    if (dataready)  
      msgSend(M1);  
    sleep(10msec);  
  }  
}
```

```
taskB( ) {  
  do { // for ever  
    msgWait(M1);  
    <process data>  
    ...  
  }  
}
```

Μετά τη δημιουργία των διεργασιών

- Μετά τη δημιουργία διεργασιών και ISRs
- Ακολουθεί ανάλυση της δυνατότητας χρονοδρομολόγησης (scheduling) στο σύστημα
 - Δυνατότητα εκτέλεσης εντός προθεσμιών
 - Με βάση τον επιλεγμένο αλγόριθμο χρονοδρομολόγησης
 - Έλεγχος χρονικής (και όχι λειτουργικής) ορθότητας
- Ο έλεγχος βασίζεται
 - Στους χρόνους εκτέλεσης των διεργασιών
 - Στην περίοδο των επαναληπτικών διεργασιών