Παράλληλη Επεξεργασία

Φροντιστήριο:

- *Συγχρονισμός (συνέχεια)*
- *Μεταβλητές υπό συνθήκη*

Εργαστήριο Πληροφοριακών Συστημάτων Υψηλής Επίδοσης

Parallel and Distributed Systems Group

Συγχρονισμός μεταξύ νημάτων

Πολύ συχνά, στις διάφορες παράλληλες εφαρμογές δημιουργείται η ανάγκη για συγχρονισμό μεταξύ των διαφόρων νημάτων

- Για την προσπέλαση από τα διάφορα νήματα κάποιας κοινής μεταβλητής
- Για να ειδοποιήσει κάποιο νήμα κάποιο άλλο όταν συμβεί κάποιο γεγονός
  - Νήματα που υλοποιούν ένα pipeline
  - Νήματα που δουλεύουν με το μοντέλο producer –consumer
- Όταν τα νήματα πρέπει να λειτουργήσουν σαν μια ομάδα
  - Όταν όλα τα νήματα πρέπει να ξεκινήσουν ταυτόχρονα να εκτελέσουν μια εργασία (π.χ. παράλληλο βρόχο)

Μεταβλητές υπό συνθήκη

- Οι μεταβλητές υπό συνθήκη παρέχουν έναν εναλλακτικό τρόπο για συγχρονισμό μεταξύ των νημάτων
- Σε αντίθεση με τις μεταβλητές αμοιβαίου αποκλεισμού, η χρήση των μεταβλητών υπό συνθήκη δεν έχει σαν σκοπό να εγγυηθεί την αποκλειστική πρόσβαση ενός νήματος σε δεδομένα (προστασία ενός κρίσιμου τμήματος)
- Οι μεταβλητές υπό συνθήκη αποτελούν ένα μηχανισμό ειδοποίησης μεταξύ των νημάτων

Μεταβλητές υπό συνθήκη

- Για να έχει νόημα η χρήση των μεταβλητών υπό συνθήκη σαν μηχανισμός ειδοποίησης θα πρέπει να συσχετίζεται, από τη μεριά του προγραμματιστή, με τον έλεγχο ορισμένων κοινών δεδομένων
- Τα κοινά αυτά δεδομένα αποτελούν και τη συνθήκη με την οποία συσχετίζεται η μεταβλητή
- Χωρίς τις μεταβλητές υπό συνθήκη, ο προγραμματιστής θα έπρεπε συνεχώς να ελέγχει (poll) την τιμή των δεδομένων, καταναλώνοντας πόρους από το σύστημα
- Αποφεύγεται το συνεχές polling στα δεδομένα, καθώς κάποιο νήμα ειδοποιεί τα ενδιαφερόμενα για την αλλαγή της τιμής των δεδομένων, μέσω μιας μεταβλητής υπό συνθήκη

Ένα παράδειγμα

- Έστω μια κοινή ουρά δεδομένων (queue)
- Ένα νήμα είναι απασχολημένο με την εξαγωγή στοιχείων από την ουρά
- Όσο η ουρά είναι άδεια το νήμα θα πρέπει να περιμένει χωρίς να εκτελεί χρήσιμη εργασία
- Στοιχεία στην ουρά τοποθετούν ορισμένα άλλα νήματα σε διάφορες στιγμές κατά την ροή του προγράμματος

Ένα παράδειγμα

- Συνθήκη: Η κατάσταση της κοινής ουράς δεδομένων. Γεμάτη ή άδεια. Ο έλεγχος μπορεί να γίνει από την μεταβλητή που αποθηκεύει το μέγεθος της ουράς
- Το νήμα περιμένει την ειδοποίηση του για την προσθήκη στοιχείου στην ουρά μέσω μιας μεταβλητής υπό συνθήκη
- Κάποια στιγμή ένα νήμα προσθέτει ένα στοιχείο στην ουρά και ειδοποιεί το νήμα που αναμένει, μέσω της μεταβλητής υπό συνθήκη
- Κοινά δεδομένα – Κρίσιμα τμήματα:
  - Η μεταβλητή μεγέθους της ουράς
  - Η κοινή ουρά δεδομένων, μέσω των πράξεων εισαγωγής, εξαγωγής στοιχείων

Δημιουργία

- Το POSIX παρέχει τον τύπο *pthread_cond_t*, οποίος ορίζει μια μεταβλητή υπό συνθήκη
- Και οι μεταβλητές αυτού του τύπου θα πρέπει να αρχικοποιηθούν πριν χρησιμοποιηθούν:
  - Στατική αρχικοποίηση:

    pthread_cond_t condition = PTHREAD_COND_INITIALIZER;

  - Δυναμική αρχικοποίηση χρησιμοποιώντας την κλήση:

    int pthread_cond_init(condition);
- Μια μεταβλητή υπό συνθήκη χρησιμοποιείται πάντα μαζί με μια μεταβλητή τύπου κλειδί

Αναμονή

- Ένα νήμα ελέγχου μπορεί να περιμένει μέχρι να τεθεί μια μεταβλητή υπό συνθήκη χρησιμοποιώντας την κλήση:

  pthread_cond_wait (condition, mutex);

- Η κλήση αυτή μπλοκάρει το νήμα που την καλεί μέχρι να τεθεί η μεταβλητή υπό συνθήκη

- Η κλήση αυτή πρέπει να καλείται με την μεταβλητή κλειδί σε κατάσταση κλειδωμένο

- Κατά την διάρκεια που το νήμα είναι μπλοκαρισμένο η μεταβλητή κλειδί απελευθερώνεται

- Όταν η κλήση αυτή επιστρέψει το κλειδί είναι πάλι σε κατάσταση κλειδωμένο

Γιατί χρειάζεται κλειδί για την αναμονή;

- Η συνθήκη, δηλαδή τα δεδομένα προς έλεγχο, είναι κοινά ανάμεσα στα νήματα (πχ το μέγεθος της ουράς)
- Η ασφαλής πρόσβαση στα κοινά δεδομένα εξασφαλίζεται με την χρήση μεταβλητών αμοιβαίου αποκλεισμού
- Άρα μια μεταβλητή υπό συνθήκη υλοποιεί την ειδοποίηση για την αλλαγή της συνθήκης και μια μεταβλητή αμοιβαίου αποκλεισμού υλοποιεί την ασφαλή/αποκλειστική πρόσβαση στη συνθήκη
- Επομένως στην κλήση της αναμονής μπορούν να συνδυαστούν και οι δυο χρήσεις των διαφορετικών μεταβλητών, εφόσον ούτως ή άλλως είναι αναγκαίες.

Αφύπνηση

- Ένα νήμα ελέγχου μπορεί να ξυπνήσει ένα νήμα που περιμένει σε κάποια μεταβλητή υπό συνθήκη με την κλήση:

  pthread_cond_signal (condition);

- Ένα νήμα ελέγχου μπορεί να ξυπνήσει όλα τα νήματα που περιμένουν σε κάποια μεταβλητή υπό συνθήκη με την κλήση:

  pthread_cond_broadcast (condition);

- Η κλήση αυτή ενδεχομένως καλείται με το σχετικό κλειδί σε κατάσταση κλειδωμένο. Μετά την κλήση αυτή το κλειδί πρέπει να ελευθερωθεί, έτσι ώστε η αντίστοιχη *pthread_cond_wait* να λειτουργήσει

Προϋποθέσεις

- Για να λειτουργήσει σωστά μια μεταβλητή υπό συνθήκη, απαιτείται κατάλληλο κλείδωμα και ξεκλείδωμα της αντίστοιχης μεταβλητής κλειδί
- Αν κληθεί η *pthread_cond_wait* χωρίς να έχει κλειδωθεί η μεταβλητή κλειδί, η συμπεριφορά της κλήσης θα είναι απρόβλεπτη
- Εφόσον έχει δεσμευθεί, αν δεν απελευθερωθεί η μεταβλητή κλειδί μετά την κλήση *pthread_cond_signal*, η αντίστοιχη *pthread_cond_wait* δεν θα μπορέσει να ξεμπλοκαριστεί

Μεταβλητές υπο συνθήκη – Παράδειγμα

Νήμα 1

```
for(...){
   ...
   pthread_mutex_lock(&mutex);

   count++;
   if(count == N){
        // Wake-up thread 2

        pthread_cond_signal(&condition);
   }

   pthread_mutex_unlock(&mutex);
   ...
   //Do work
}
```

Νήμα 2

```
...
pthread_mutex_lock(&mutex);

while(count<N) {
   pthread_cond_wait(&condition, &mutex);

}
count -= N;
pthread_mutex_unlock(&mutex);
...

//Do work
```

Σημαντικό

- Πάντα πρέπει να γίνεται επανέλεγχος της συνθήκης μετά από το ξεμπλοκάρισμα από την αναμονή σε μια κλήση της *pthread_cond_wait*
- Αυτό μπορεί να γίνει τοποθετώντας την *pthread_cond_wait* στο εσωτερικό ενός while loop στο οποίο θα ελέγχετε η συνθήκη
- Αυτό είναι ιδιαίτερα σημαντικό διότι:
  - τα νήματα είναι ασύγχρονα. Κάποιο άλλο νήμα μπορεί να έχει προλάβει να μεταβάλει την συνθήκη
  - Υπάρχει η (σπάνια – αλλά όχι και τόσο σπάνια) περίπτωση των συμπτωματικών ειδοποιήσεων (spurious wakeups). Ειδοποιήσεις οι οποίες προκαλούνται χωρίς να έχει προηγηθεί κάποιο signal ή broadcast της συγκεκριμένης μεταβλητής υπό συνθήκη από άλλο νήμα

- *Απορίες και συζήτηση μέσω του forum του μαθήματος στο My.CEID*
- *e-mail επικοινωνίας kik@hpclab.ceid.upatras.gr*
- *Ανακοίνωση ασκήσεων και υλικό στην ιστοσελίδα του μαθήματος http://parallel.hpclab.ceid.upatras.gr/*