

Άσκηση 1η: Υπολογισμός αθροίσματος με χρήση συνεργαζόμενων διεργασιών (νημάτων)

Υλοποίηση 1: Με διεργασίες

Το πρόγραμμα υπολογίζει το άθροισμα $1+...+N$ χρησιμοποιώντας n διεργασίες, οι οποίες υπολογίζουν το μερικό άθροισμα N/n όρων. Αποτελείται από 3 συναρτήσεις τη `main()`, τη `child_creator()` και την `child_sum()`. Αναλυτικά οι λειτουργίες σε κάθε συνάρτηση:

Στη `main()`

- Ορίζονται οι μεταβλητές N, n που δίνει ο χρήστης από τη γραμμή εντολών
- Δημιουργείται το `pipe` για την διαδιεργασιακή επικοινωνία μεταξύ του γονέα και των παιδιών
- Ο γονέας καλεί την `child_creator()` και όταν αυτή επιστρέψει, η γονική διεργασία διαβάζει από το `pipe` τα μερικά αθροίσματα που έχουν υπολογίσει τα παιδιά του. Υπολογίζει το τελικό άθροισμα και το συγκρίνει με αυτό που δίνεται από τον τύπο $N*(N+1)/2$

Στη `child_creator()`:

- Δημιουργούνται επαναληπτικά οι διεργασίες παιδιά με τη `fork()`. Κάθε μία καλεί την `child_sum()` και γράφει το μερικό της άθροισμα στο `pipe`.
- Η γονική διεργασία με τη `wait(NULL)` περιμένει όλες τις διεργασίες παιδιά της να τερματιστούν και μετά τερματίζει

Στη `child_sum()`:

- Υπολογίζεται και επιστρέφεται το μερικό άθροισμα της κάθε διεργασίας

Παρουσιάζονται τα αποτελέσματα της εκτέλεσης

NATALIA ΡΟΥΣΚΑ

1092581

10/11/2024

```
natalia@debian:~/Source$ gcc myprocesses.c
natalia@debian:~/Source$ ./a.out 1000 10
Count the sum of N=1000 integers using n=10 processes
Hello from child 0 with sum 5050
Hello from child 1 with sum 15050
Hello from child 2 with sum 25050
Hello from child 3 with sum 35050
Hello from child 4 with sum 45050
Hello from child 5 with sum 55050
Hello from child 6 with sum 65050
Hello from child 7 with sum 75050
Hello from child 8 with sum 85050
Hello from child 9 with sum 95050
The sum counted by the parent reading the pipe is 500500
The sum counted by the parent is 500500
natalia@debian:~/Source$
```

Υλοποίηση 2: Με νήματα

Το πρόγραμμα υπολογίζει το άθροισμα $1+...+N$ χρησιμοποιώντας n νήματα, τα οποία υπολογίζουν το μερικό άθροισμα N/n όρων. Αποτελείται από 3 συναρτήσεις τη `main()`, τη `thread_creator()` και την `thread_count_sum()`. Αναλυτικά οι λειτουργίες σε κάθε συνάρτηση:

Στη `main()`

- Ορίζονται οι μεταβλητές N, n που δίνει ο χρήστης από τη γραμμή εντολών
- Δημιουργείται ένας πίνακας `mythread_data_array` με n στοιχεία τύπου `thread_data`
- Ο γονέας καλεί την `thread_creator()` και όταν αυτή επιστρέψει, η γονική διεργασία αθροίζει τα μερικά αθροίσματα που έχουν υπολογίσει τα νήματα που δημιουργήθηκαν, διαβάζοντας το πεδίο `sum` του κάθε στοιχείου του πίνακα. Υπολογίζει το τελικό άθροισμα και το συγκρίνει με αυτό που δίνεται από τον τύπο $N*(N+1)/2$

Στη `thread_creator()`:

- Δημιουργούνται επαναληπτικά τα νήματα με την `pthread_create(pthread_t *th, const pthread_attr_t *attr, void *(*start_routine)(void *), void* arg)`, η οποία παίρνει 4 ορίσματα. Το πρώτο είναι δείκτης στο `id` του νήματος, το δεύτερο ορίζεται με `NULL` για default attributes, το τρίτο είναι δείκτης στη συνάρτηση που θα εκτελεστεί από το νήμα (`thread_count_sum`) και το τελευταίο το όρισμα της συνάρτησης `thread_count_sum`. Επιτρέπει μόνο ένα όρισμα για τη συνάρτηση που θα εκτελεστεί οπότε ορίζουμε μία δομή `thread_data` και περνάω σαν όρισμα την διεύθυνση του κάθε στοιχείου τύπου `thread_data` του πίνακα `mythread_data_array`.
- Καλείται η `pthread_join()` επαναληπτικά για να τερματήσουν όλα τα νήματα πριν συνεχιστεί το κύριο νήμα

Στη `thread_count_sum()`:

NATALIA ΡΟΥΣΚΑ

1092581

10/11/2024

- Υπολογίζεται το μερικό άθροισμα για το τμήμα που αντιστοιχεί στο νήμα που την καλεί, δηλαδή από start έως end και αποθηκεύεται στο πεδίο sum της δομής thread_data που είχε ως όρισμα

Παρουσιάζονται τα αποτελέσματα της εκτέλεσης

```
natalia@debian:~$ cd Source
natalia@debian:~/Source$ ls
a.out  mythreads.c
natalia@debian:~/Source$ gcc -pthread mythreads.c
natalia@debian:~/Source$ ./a.out 1000 10
Count the sum of N=1000 integers using n=10 threads
Hello from thread 2 with sum 25050
Hello from thread 3 with sum 35050
Hello from thread 4 with sum 45050
Hello from thread 5 with sum 55050
Hello from thread 6 with sum 65050
Hello from thread 7 with sum 75050
Hello from thread 8 with sum 85050
Hello from thread 9 with sum 95050
Hello from thread 1 with sum 15050
Hello from thread 0 with sum 5050
The sum counted by the threads is 500500
The sum counted by the parent is 500500
```