InfoLab

IC621: Distributed and Parallel Computing

Lab 05: Linked List with Fine-Grained Locking

Min-Soo Kim

DGIST 대구경북과학기술원
Daegu Gyeongbuk Institute of Science & Technology

Implement Insert()

- **Understand how to implement the Member() function of fine-grained locking**
  - from the Member() function of coarse-grained locking
- **Implement the Insert() function of fine-grained locking, especially using pthread's mutex (not read-write locks)**
- **Write a main function for testing (creating four threads)**
  - start with an empty linked list (created by main thread)
  - thread 1 calls Insert() with the values {1, 5, 9, …, 97}
  - …
  - thread 4 calls Insert() with the values {4, 8, 12, …, 100}
- **Write a function for printing the linked list and use it for showing the correctness of your implementation**

```
1  int  Member(int value, struct list_node_s* head_p) {
2     struct list_node_s* curr_p = head_p;
3
4     while (curr_p != NULL && curr_p->data < value)
5        curr_p = curr_p->next;
6
7     if (curr_p == NULL || curr_p->data > value) {
8        return 0;
9     } else {
10       return 1;
11    }
12 }  /* Member */
```

```
int  Member(int value) {
   struct list_node_s* temp_p;

   pthread_mutex_lock(&head_p_mutex);
   temp_p = head_p;
   while (temp_p != NULL && temp_p->data < value) {
      if (temp_p->next != NULL)
         pthread_mutex_lock(&(temp_p->next->mutex));
      if (temp_p == head_p)
         pthread_mutex_unlock(&head_p_mutex);
      pthread_mutex_unlock(&(temp_p->mutex));
      temp_p = temp_p->next;
   }

   if (temp_p == NULL || temp_p->data > value) {
      if (temp_p == head_p)
         pthread_mutex_unlock(&head_p_mutex);
      if (temp_p != NULL)
         pthread_mutex_unlock(&(temp_p->mutex));
      return 0;
   } else {
      if (temp_p == head_p)
         pthread_mutex_unlock(&head_p_mutex);
      pthread_mutex_unlock(&(temp_p->mutex));
      return 1;
   }
}  /* Member */
```

InfoLab

Thank you!

DGIST 대구경북과학기술원
Daegu Gyeongbuk Institute of Science & Technology