

Soru 1. Koşut işlem kapsamında “hızlanma,” “etkinlik” ve “ölçeklenebilirlik” kavramlarını açıklayınız.

Soru 2. “Paylaşımlı Bellekli MIMD” mimaride, MPI işlevlerini kullanan ve “Dağıtılmış Bellekli MIMD” için geliştirilmiş yazılımları çalıştırmak için bir gereksinim olduğunu varsayınız. “MPI\_Barrier” işlevini “Paylaşımlı Bellekli MIMD” mimaride gerçekleştirmek için:

- Tasarım yapınız,
- İşlevleri C Programlama Diliyle kodlayınız.

<pre>int MPI Barrier(MPI Comm comm)</pre>
<p><i>The only argument of MPI_Barrier is the communicator that defines the group of processes that are synchronized. The call to MPI_Barrier returns only after all the processes in the group have called this function.</i></p>

### Zaman uyumlama işlevlerini içeren örnek program

```
1 pthread_cond_t cond_queue_empty, cond_queue_full;
2 pthread_mutex_t task_queue_cond_lock;
3 int task_available;
4
5 /* other data structures here */
6
7 main() {
8     /* declarations and initializations */
9     task_available = 0;
10    pthread_init();
11    pthread_cond_init(&cond_queue_empty, NULL);
12    pthread_cond_init(&cond_queue_full, NULL);
13    pthread_mutex_init(&task_queue_cond_lock, NULL);
14    /* create and join producer and consumer threads */
15 }
16
17 void *producer(void *producer_thread_data) {
18     int inserted;
19     while (!done()) {
20         create_task();
21         pthread_mutex_lock(&task_queue_cond_lock);
22         while (task_available == 1)
23             pthread_cond_wait(&cond_queue_empty,
24                               &task_queue_cond_lock);
25         insert_into_queue();
26         task_available = 1;
27         pthread_cond_signal(&cond_queue_full);
28         pthread_mutex_unlock(&task_queue_cond_lock);
29     }
30 }
```