

İşletim Sistemleri 2. Proje Raporu

Projenin Adı: C, Unix/Linux Multithreading, Pthreads, Process Synchronization

Projenin Konusu: Meşhur Dining Philosophers problemini,lokantaya gelen her filozof 1 thread olacak şekilde çalışmasını sağlayıp, filozofların yediklerini pirinç miktarına göre masa hesaplarını gösteren bir program yazmak.

Tamamlanan Talimatlar:

- Gelecek Filozof Miktarını input olarak almasını sağladım.
- Gelecek filozof miktarına göre filozofları 8li masalara oturttum.Hesap ve masa oturushlarında masa sayısını doğru olmasını sağladım.
- Gelecek filozofları masalara oturttuğumda tam dolmayan masadaki filozofların da yeme düşünme döngülerini hazırladım
- Eğer kullanıcı gelecek filozof miktarını lokantanın kapasitesi olan seksen sayısından fazla olursa,o parametreyi kabul etmeyip kullanıcıdan tekrar gelecek filozof miktarı alan bir döngü kodu hazırladım.
- Kullanıcının girdiği gelecek filozof miktarı parametresi ile döngü oluşturdum.Bu döngünün içinde filozofların düşünme yeme döngüsünü temsil eden **yeme()** kodu oluşturdum.Her filozofun bir thread olması için bir **pthread_t** arrayi oluşturdum. Arrayin boyutu olarak 80 olarak belirledim çünkü 10 masa ve her masa 8 kişi olduğu için lokantanın limiti 80 olduğu için.
- Yukarıda yazmış olduğım filozof yeme döngüsü olan **yeme()** fonksiyonunu biraz daha açıklamak istiyorum.Bir filozof threadi bu fonksiyonu çalıştırdığında rastgele olayı için bu kodu **srand((unsigned) time(&t));** yazdım.Bu kod sayesinde kullandım bütün rand() fonksiyonlarının verdiği rakamlar farklı olmasını sağladım. **(rand()%5)+1;** komutu sayesinde 1-5 arasında rastgele bir rakam oluşturunuyorum. 2 defa rastgele değer alıyoruz.Aldığımız değerlerden biri kaç yüz gram pirinç yediği ile ilgiliyken diğer değer kaç milisaniyede yemeği yediği ile ilgili.Yeme süresi miktarıyla kodu bekletiyorum.Bunun için **usleep()** kodu kullandım.Filozofların yediği pirinç miktarı genel bir değışkende birikiyor, sonraki masaya geçtiğinde ise sıfırlanıyor.

Gerçekleştirilemeyenler ve Sınırlandırmalar:

-**yeme()** fonksiyonunda rastgele olması için rand() ve srand() fonksiyonlarını kullandığımı yukarıda bahsetmiştim.Bu kısmı biraz açıklamak istiyorum.C'de rand() fonksiyonu size döndürdüğü değer sabittir.Bu değeri değıştirmek için srand() fonksiyonunu kullanırız.İnternette bu değerin rastgele olması için bir çok srand parametresi gördüm.Hepsini denesemde kesin anlamda her çalıştığında

rastgele deęer gelen bir kod bulamadım.Bu hatanın temel sebebi bana verilen rastgele deęerin 5e gre modunu alıp 1 artırdıęımdan, kk bir aralıktaki kaldıęından olabilir.

-usleep() fonksiyonu bazen aldıęımız deęerleri bozuyordu.Ben de masadaki herkes yedikten sonra hesap gelmeden 1 saniye sleep() fonksiyonu ile bekletiyorum

Thread Kullanmanın Avantajları:

Bu projede senkronize threadler kullandık.Threadler fork yapılmıř bir prosese gre ok daha hafiftir .nk prosesin tamamen kopyalanması gerekmez ve daha hızlı iřlemler yapar.Bir proses iin birden fazla thread kullanımı iřlem performansını artıracaktır.nk threadler oluřturulduęu prosesin kaynaęına eriřerek kendileri arasında da iyi bir iletiřim gerekleřtirmiř olurlar.

Multithread Senkronizasyonu:

Birden Fazla thread kullanmak iř ykn hafifletip performans saęlasa da iyi bir senkronizasyon istemektedir.Kritik blgeri belirleyip hangi prosesin iřlem yapması gerekiyorsa ona gre lock mekanizmalarımızı oluřturmaliyiz.

Yardıma aldıęım kaynaklar:

-labarotuar kaynakları ve rnek programlar

-Code Vault kanalının “Unix Threads in C” oynatma listesi

-stackoverflow

- https://www.tutorialspoint.com/c_standard_library/c_function_rand.htm

Ahmet Yanık

1821221001

