

BLG 312– Bilgisayar İşletim Sistemleri

2019–2020, Ödev 2

Giriş

Bu ödevde sizden çoklu-proses ve çoklu-iplik(thread) kullanarak asal sayıları bulan bir program yazmanız beklenmektedir. Arama yapılacak sayı aralığı(**interval_min** and **interval_max**), proses sayısı(**np**) ve iplik sayısı(**nt**) komut satırı argümanları olarak programınıza verilecektir. Verilecek sayı aralığı, arama yapılacak aralığın kısıtlanmasında kullanılacaktır. Başlangıçta bu aralığın **np** sayıda proses'e bölünmesi ve ardından her prosesin de kendisine düşen aralığı **nt** sayıda ipliğe bölmesi gerekmektedir. İplikler verilen aralıkta asal sayıları bulacaktır. Proses içerisinde oluşturulmuş ipliklerin çalışması tamamlandığında, prosesler bulunan asal sayıların bilgisini paylaşılan bellek üzerinden ana prosese aktarması gerekmektedir. Son olarak, ana prosese bulunan tüm asal sayılarını küçükten büyüğe sıralı olarak ekrana yazdıracaktır. Gerçeklemenizi tek kaynak kodu içerisinde yapıp ninovaya yüklemeniz gerekmektedir.

Program Girdisi

Program girdisi olarak: sayı aralığı (interval_min, interval_max), proses sayısı(**np**) ve iplik sayısı(**nt**) verilecektir. Bu girişler komut satırından okunmalıdır. Programlarınız değerlendirilirken farklı girdiler verilerek programınız test edilecektir.

./yourprogram interval_min interval_max np nt

Örnek Komut: **./yourprogram 101 200 2 2**

2 proses yaratılır ve her proses 2 iplik yaratır.

Proses 1 şu aralıktan sorumludur: **101-150**

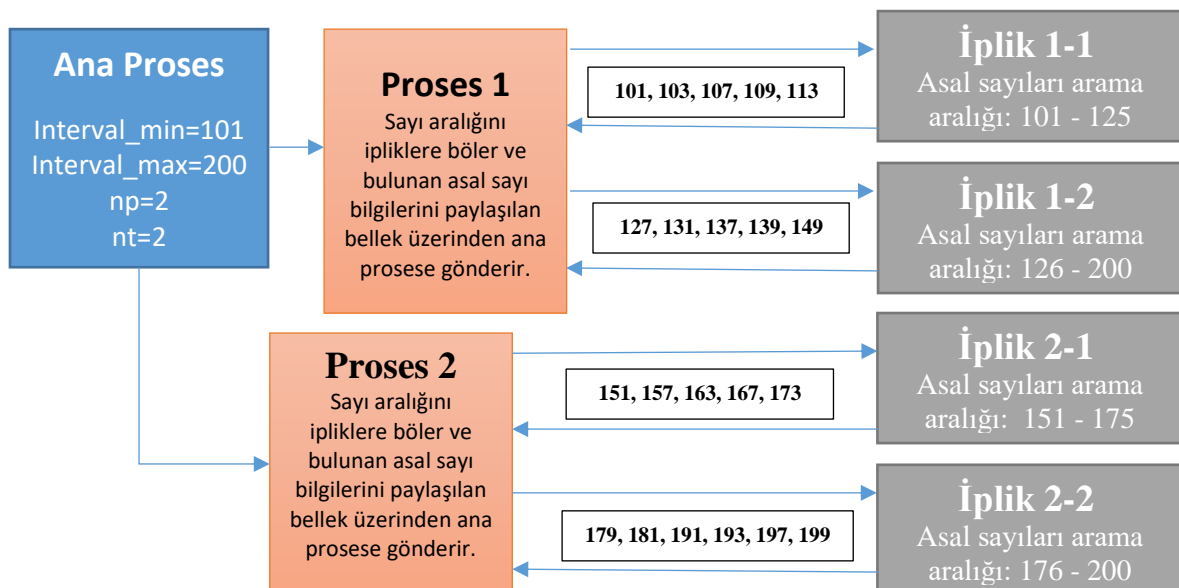
İplik 1-1 şu aralıkta asal sayı testi yapacaktır: **101-125**

İplik 1-2 şu aralıkta asal sayı testi yapacaktır: **125-150**

Proses 2 şu aralıktan sorumludur: **151-200**

İplik 2-1 şu aralıkta asal sayı testi yapacaktır: **151-175**

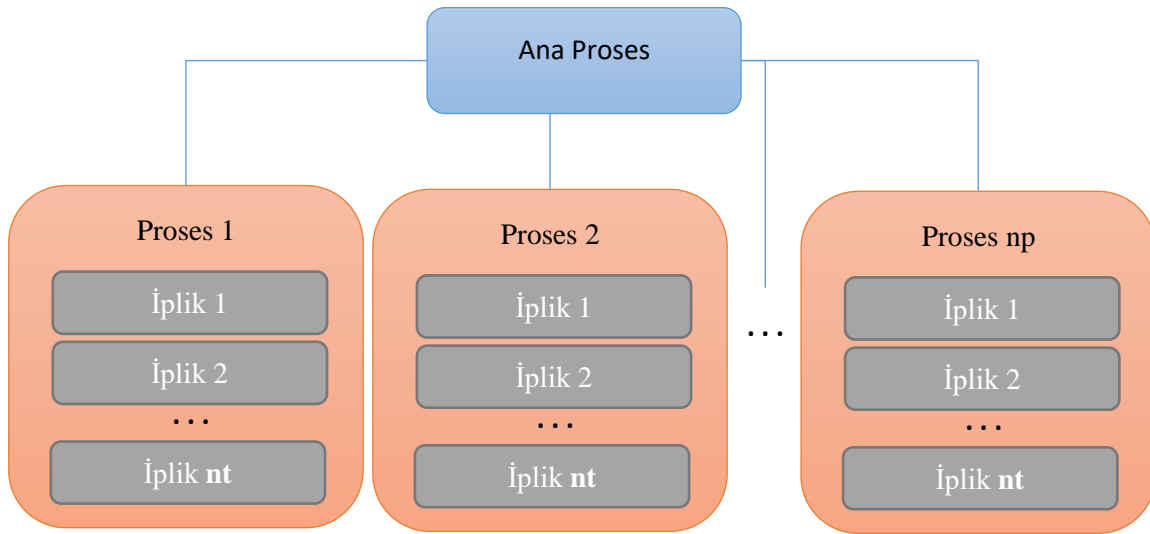
İplik2-2 şu aralıkta asal sayı testi yapacaktır: **176-200**



Prosesler ve İplikler

Yazacağınız programda proses ve iplikler için aşağıda belirtilen özelliklerin bulunmalıdır:

- **Ana Proses:** Ana prosesiniz gerekli argümanları komut satırı argümanından okumalıdır. Ardından **np** sayıda proses oluşturmalıdır. Bununla beraber her bir prosesin kullanacağı sayı aralığını da belirlemeli ve prosesler arasında bölüştürmelidir. (Örneğin: Eğer başlangıçta verilen aralık 1-10 ve proses sayısı(**np**) 2 ise; prosesler 1-5 ve 6-10 aralığına atanmalıdır.). Oluşturulan proseslerin çalışması tamamlandığında, ana proses **paylaşılan bellek** üzerinden gönderilmiş asal sayıları **artan** sırada ekrana bastırmalıdır.
- **Çocuk prosesler:** Tüm çocuk proseslerin **nt** sayıda iplik(thread) üretmesi ve bu ipliklere bir sayı aralığı ataması gerekmektedir. İpliklerin çalışması tamamlandığında, çocuk prosesler iplikleri tarafından hesaplanan asal sayı bilgilerini, paylaşılan bellek üzerinden ana prosese göndereceklerdir.
- **İplikler:** Her iplik, atanan sayı aralığında bulunan sayılara asal sayı testi yapacaktır. Asal sayı testi için basitçe bölme testi kullanılması yeterlidir.
- **Her proses ve iplik örnek çıktıda görüldüğü şekilde başlangıç ve bitiş çıktılarını ekrana bastırmalıdır.**
- **Her proses ve iplik paralel bir şekilde çalışabilmelidir.**



Geliştirme ortamı bilgilerini, derleme ve çalıştırma komutlarını kaynak dosyanızın başına yorum satırları içerisinde eklemelisiniz.

Örnek:

```
// Geliştirme ortamı: Ubuntu 16.04 or ITU SSH servers
// Derleme komutu: g++ -c Homework1.cpp -pthread
// Çalıştırma komutu: ./a.out interval_min interval_max np nt
// Örnek: ./a.out 101 200 2 2
```

Örnek Program Çıktısı

Aşağıda örnek bir program çıktısı verilmiştir. Oluşturacağınız programın benzer bir şekilde çıktı üretmesi gerekmektedir. **Dikkat: Her çalışmada proseslerin ya da ipliklerin çalışma sıraları farklılık gösterebilir.**

```
./yourprogram 101 200 2 2
```

Ana Proses: basladi.

Proses 2: basladi. Aralik: 151-200

İplik 2.2: araniyor: 176-200

Proses 1: basladi. Aralik: 101-150

İplik 1.2: araniyor. Aralik: 126-150

İplik 1.1: araniyor. Aralik: 101-125

İplik 2.1: araniyor. Aralik: 151-175

Proses 2: Sonlandi.

Proses 1: Sonlandi.

Ana Proses: Sonlandi. Bulunan asal sayilar: 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199,

Önemli notlar:

- Bu ödev üzerinde **bireysel olarak çalışmanız** beklenmektedir. Ödev çözümlerinde yapılacak işbirliği yapmamanız beklenmekte ve tespit edildiği takdirde kopya olarak değerlendirilecektir. Bir başkasının çalışmasını kendi çalışmanız olarak göndermek (bir parçası olsa ve üzerinde değişiklik yapılmış olsa da) ya da farklı kaynaklardan (internet de dahil olmak üzere) atıf yapılsa bile kopyala / yapıştır yaparak kod ya da metin içeriklerini ödevinizde kullanmanız gibi eylemler de bu kapsama girmektedir. Tespit edildiği takdirde ilgili durum disiplin soruşturması için idareye bildirilecektir. Kopya eylemine katılan tüm taraflar eşit şekilde muamele görecektir.
- Ödevinizi Ninova sistemi üzerinden tek dosya olarak (kaynak kodu) teslim etmeniz gerekmektedir. Geç teslimler kabul edilmemektedir.
- Tüm kodunuz C programlama dili ile yazılmalı ve gcc kullanarak ITU'nun Linux Sunucusu'nda (SSH üzerinden erişebilirsiniz) derlenebilmeli ve çalıştırılabilir. Kodunuzun hatasız derlenmesi beklenmektedir; aksi takdirde, ödevin notu sıfır olarak değerlendirilebilir.
- Yazdığınızın **programı derlemek ve çalıştırmak için gerekli talimatları**, yorum satırları olarak kodunuzun başlangıcına eklemeniz gerekmektedir.
- Ödevle ilgili sorularınız için, ders yardımcısı Çağatay KOÇ (kocca@itu.edu.tr) ile iletişime geçebilirsiniz.

Ek

Aşağıda bulunan kod örneklerinden komut satırı argümanlarını okumak ve karakter dizilerini sayılara çevirmek için faydalanabilirsiniz.

- https://rosettacode.org/wiki/Command-line_arguments
- <https://www.geeksforgeeks.org/convertng-strings-numbers-cc/>