

INF 443
Dağıtık Sistemler ve Uygulamaları
Ödev III

Serhan Daniş

13.10.2017

İçindekiler

1	Log yazma	1
2	π'ye yakınsama	1

1 Log yazma

- Thread şeklinde çalışan bir kayıt tutucu yazacağız. Kaydetmek istediğimiz bilgileri bir kuyruğa yazılacak. Bu kuyruğu okumakla sorumlu olan bir kayıt tutucu thread olacak. Kuyrukta bir bilgi varsa bunu başına zaman bilgisini de koyarak bir dosyaya ekleyecek.

2 Paralel programlamayla π 'ye yakınsama

π sayısına yakınsayabilmek için Nilakantha metodunu kullanacağız. Bu metod aşağıdaki seri ile tanımlanır.

$$\begin{aligned} P &= 3 + 4 \sum_{k=0}^N \frac{-1^k}{(2k+2)(2k+3)(2k+4)} \\ &= 3 + 4 \left(\frac{1}{2.3.4} - \frac{1}{4.5.6} + \frac{1}{6.7.8} - \frac{1}{8.9.10} \dots \right) \\ \lim_{N \rightarrow \infty} P &= \pi \end{aligned}$$

- N sayısını parametre olarak alan ve bir döngü kullanarak π sayısını hesaplayabilen bir fonksiyon yazın. Fonksiyonda bütün sayı değişkenleri için

Decimal paketini kullanın. Böylelikle π sayısının daha çok basamağını görebileceğiz.

```
from decimal import *
getcontext().prec = 100
sum = Decimal(3)
```

- Şimdi fonksiyonu sadece verilen k için hesaplama yapabilecek şekle dönüştürün.
- Yeni bir **alt süreç** (subprocess) için bir fonksiyon yazacağız. Bu fonksiyon ya da süreç sonsuz döngü içinde çalışıyor olup kendisine atanmış bir kuyruğu devamlı olarak kontrol ediyor olacak. Bu kuyruğa “QUIT” gönderilirse döngüyü kesecek, bir sayı aralığı verilirse bu sayı aralığında Nilakantha metodu hesaplamasını yapıp çıktı olarak verilmiş başka bir kuyruğa (outQueue) cevabı yazacak ve tekrar beklemeye başlayacak.
- Alt süreçler açılır ve kapanırken log kuyruğuna durum bilgilendirmesi yapacak.
- **Ana thread**, yani bütün alt süreçleri ve threadleri çağıran bölüm, süreç ve threadleri çağırdıktan sonra hesaplanacak sayıları içeren ilgili mesajları alt süreçlerine (işçi süreçleri) gönderip hesaplama yapmalarını bekleyecektir. İşçiler hesaplamalarını yine bir kuyruğa yazdıklarında bu kuyruktan sonuçları toplayıp birleştirip logQueue’ya gönderecektir.
- Hesaplama bittiğinde bütün süreçlere ve threadlere düzgün olarak kapanmaları için gerekli komutları gönderecek ve kendisi de sonlanacaktır.