



PAMUKKALE ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

PROGRAMLAMA DİLLERİ

ÖDEVİ

BEYZA KAYNAR

17253022

DENİZLİ, MAYIS 2019

Chapel Programlama Dili



Chapel, Cray Inc. tarafından ortaya konmuş bir paralel programlama dilidir. Açılımı “Cascade High Productivity Language”tır. İlk olarak 2010 yılında versiyon 1.1 yayınlanmıştır. Şu an ise 5 Nisan 2019 yılında hazırlanan 1.19 versiyonu yayındadır.

Chapel dili büyük ölçekli sistemlerde, paralel programlama için kolaylık sağlaması amacıyla dizayn edilmiştir. Ve çok çekirdekli bilgisayarlarda, bulutlarda ve yüksek kaliteli süper bilgisayarlarda çalışacak şekilde tasarlanmıştır.

Chapel dilinin en temel amacı cascade kontrol sistemlerinde paralel programlama yapabilmektir. (Paralel Programlama: Bir problemin çözülürken ufak görev parçalarına ayrıştırılıp ve bu görev parçalarının eşzamanlı olarak sistemli bir şekilde çalıştırılması yöntemidir. Bu yöntem sayesinde performans artışı olmakta ve var olan büyük problemler daha kısa sürede çözülmektedir.)

Örnek:

```
begin writeln("hello world");  
writeln("goodbye");
```

Yukarıdaki kodun olası çıktıları:

```
hello world  
goodbye
```

```
goodbye  
hello world
```

Bölmeli global adres aralığı, nesneye yönelik programlama ve paralel programlama paradigmalarını desteklemektedir. Veri dağıtımı için soyutlamalar ve alt hesaplamaları veriye dayalı yerleştirme yoluyla programdaki veri ve hesaplama yerinin optimizasyonunu sağlamaktadır. Nesneye yönelik kavramlar ve genel programlama özellikleri sayesinde kodun yeniden kullanılmasına ve genellenmesine izin verir.

Chapel dilinin bazı temel özellikleri:

- Overloading
- Argument intents
- Önceden tanımlanmış olan değerler (default values)
- Match-by-name
- Meta programlama için “Derleme zamanı” özellikleri
- Bağımsız Kademeli (rank-independent) programlama özellikleri

Syntax Özellikleri

Chapel dilinin syntax’ı C diline benzemektedir. Aynı C dilindeki gibi komutlar süslü parantez içerisine yazılmakta, satır sonları noktalı virgülle ifade edilmektedir. Koşullu ifadeler ve while döngüsü de C’deki yazıma çok benzemektedir. Chapel dilinde basit tipler; boolean değeri (bool), integer (int), gerçek ve sanal kayan nokta değerleri (real ve imag), karmaşık değerler (complex) ve stringlerdir (string). Yorum satırları da çift backslash (/) ile gösterilmektedir. Sınıflar “class” anahtar kelimesiyle ifade edilir. Chapel, çok boyutlu arrayler ve rank-independent programlama özelliği için genellikle tuple kullanmaktadır. (Tuple: Veri tabanında kayıtları oluşturan veri grubu)

Örnek:

- **var** t: (**int**, **real**) = (1, 2.3) ; // bir int bir de real bulunun t isimli bir tuple
- **var** (j, k) = t; // j ve k değişkenlerine sahip bir de-tuple
- ...t(1)... // t’nin birinci bileşenini yollamaktadır
- **var** coord: (**real**, **real**, **real**); // 3 tane aynı tipte tuple tanımlandı
- **var** coord: 3*real; // bir üstteki kodun farklı şekilde yazılmış hali

Chapel dilinde aralıklar “1..n” bu şekilde gösterilmektedir. Yani “0..” ifadesi negatif olmayan integerların tümü anlamına gelmektedir. Diziler (array) ise köşeli parantezlerle ([]) gösterilmektedir. Aralıklar ve arrayler üzerinde yapılan işlemleri örneklerle ele alalım.

Örnek:

- 1..9 // 1, 2, 3, 4, 5, 6, 7, 8, 9
- 1..9 **by** 2 // 1, 3, 5, 7, 9
- 1..9 **by** -1 // 9, 8, 7, 6, 5, 4, 3, 2, 1
- 1..9 **#** 3 // 1, 2, 3
- 1..9 **#** -3 // 7, 8, 9
- (1..9) [6.. **by** 2] // 6, 8
- 0..**#**n // 0, 1, 2, n-1
- **var** Hist: [-3..3] **int**; // -3, 3 aralığındaki integerlardan oluşan array
- **var** Mat: [0..**#**n, 0..**#**n] **complex**; // 0, n-1 aralığında complex sayılardan oluşan çift boyutlu array
- **var** Tri [i in 1..n] [1..**i**] **real**; // Reel sayılardan oluşan üçgen şeklinde array

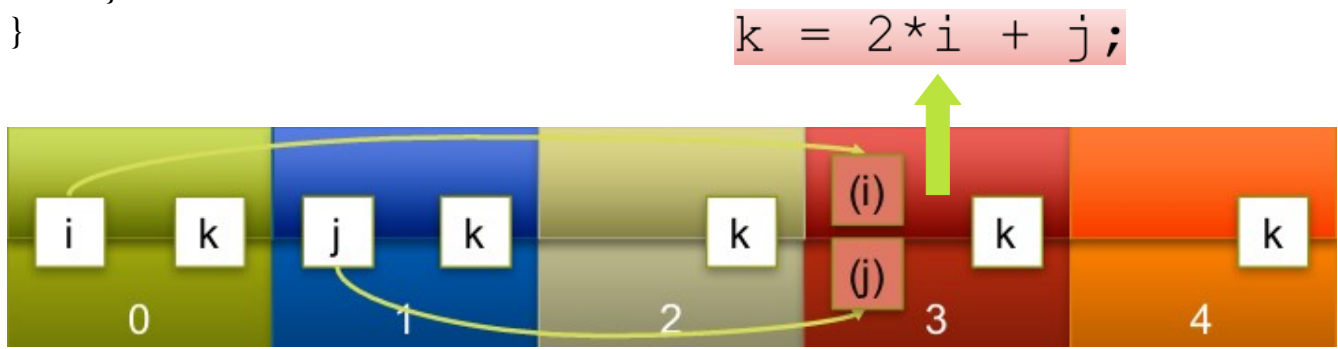
Chapel’in for döngüsü C diliden farklı ifade edilmektedir. İterasyon değişkeninin aralık içinde gezinmesi şeklinde yazılır.

Örnek:

- **for** k in 1..n **do** // çıktı olarak “1 2 3 4 5 ... n” verir
 write (k);

Scoping ve Locality

```
var i : int;  
on Locales [1] {  
  var j : int;  
  forall loc in Locales {  
    on loc {  
      var k : int;  
      k = 2*i+j;  
    }  
  }  
}
```



Farklı Merhaba Dünya Uygulamaları

- `writeln ("Hello World!");`
- ```
module Hello {
 config const message = "Hello World!";
 proc main () {
 writeln (message);
 }
}
```
- `//Parallel`
- ```
config const n = 1000;  
forall i in 1..n do  
  writeln ("Hello World #", i) ;
```

İkili Arama

```
proc binsearch (A: [], value) {  
    var low = A.domain.dim(1).low;  
    var high = A.domain.dim(1).high;  
  
    while (low <= high) {  
        var mid = (low + high) / 2;  
  
        if A(mid) > value then  
            high = mid - 1;  
        else if A(mid) < value then  
            low = mid + 1;  
        else  
            return mid;  
        }  
    return 0;  
}
```

Faktöriyel Hesabı

```
proc fac (n) {  
    var r = 1;  
    for j in 1...n do  
        r *= j;  
    return r;  
}
```

Fibonacci Dizisi

```
iter fib (n) {  
    var current = 0;  
    next = 1;  
    for k in 1..n {  
        yield current;  
        current += next;  
        current <=> next;  
    }  
}
```

Matris Çarpımı

```
proc *(a:[], b:[]) {  
  
    if (a.elType != b.elType) then  
        writeln("type mismatch: ", a.elType, " ", b.elType);  
  
    var ad = a.domain.dims();  
    var bd = b.domain.dims();  
    var (arrows, acols) = ad;  
    var (brows, bcols) = bd;  
    if (arrows != bcols) then  
        writeln("dimension mismatch: ", ad, " ", bd);  
  
    var c: [{arows, bcols}] a.elType = 0;  
  
    for i in arows do  
        for j in bcols do  
            for k in acols do  
                c (i,j) += a (i,k) * b( k,j);  
            return c;  
        }  
    }
```

KAYNAKÇA:

- <https://chapel-lang.org/>
- <https://www.quora.com/What-is-the-Chapel-parallel-programming-language>
- [https://codesnippets.fandom.com/wiki/Chapel_\(programming_language\)](https://codesnippets.fandom.com/wiki/Chapel_(programming_language))
- <https://github.com/chapel-lang/>
- http://upc.gwu.edu/tutorials/tutorials_sc2003.pdf
- <https://chapel-lang.org/presentations/ChapelForATPESC2016-presented.pdf>
- <https://chapel-lang.org/publications/PMfPC-Chapel.pdf>