10.12.2018

BLM2031 YAPISAL PROGRAMLAMA - DÖNEM PROJESI

PSEUDO KOD DÖNÜŞTÜRÜCÜSÜ

Projenin Amacı ve Kapsamı

Gerçekleştireceğiniz projenin temel amacı pseudo kodun yazılı olduğu bir text dosyasını, belli kuralları takip ederek bir c kaynak kodu dosyası haline getirmeniz ve bu c kaynak kodunu derleyerek exe dosyasını üretmenizdir.

C kaynak koduna çevirelecek pseudo kod aşağıdaki kurallara göre oluşturulmuş olacaktır:

Aşağıda listelenen **reserved word'ler** (ör: int, float, for, while, if, begin, procedure, vb..) dışındaki ifadeler, bir değişkene karşılık gelmektedir.

Değişken olduğunu tespit ettiğiniz bir ifadede isimlendirme;

- _c ile sonlandırılmışsa, ilgili değişken char,
- _f ile sonlandırılmışsa, ilgili değişken float,
- _i ile sonlandırılmışsa, ilgili değişken integer,
- _l ile sonlandırılmışsa, ilgili değişken long,
- _d ile sonlandırılmışsa, ilgili değişken double,
- _ld ile sonlandırılmışsa, ilgili değişken long double,
- _file ile sonlandırılmışsa, ilgili değişken file pointer tipindedir.
- Diğer pointer veri türleri için: _cp: char *, _fp: float*, _ip: int*, _lp: long pointer*, _dp:double*, _ld: long double pointer* tiplerine karşılık gelmektedir.
- Dizi indislerine erişim için C'deki gibi köşeli parantezler [] kullanılır, dizinin veri tipi yukarıda yazılan kurala göre belirlenir (ör: x_f[i_i]: x isminde, float tipte bir dizinin i'nci adresine erişim, dikkat: i indisi de integer tipinde bir değişkendir).
- Tüm diziler ve stringler için eleman sayısı (aksi belirtilmediği sürece) default olarak 100'dür.

  Açıkça aşağıdaki gibi bir tanım yapıldıysa (yapılmak zorunda değil), dizi farklı bir boyuta sahip olabilir;
  DEFINE arr_c[10] : arr, 10 elemanlı bir character dizisidir.
- Stringler stringIsmı_c[] şeklinde gösterilir.
  Ör: str1 ve str2'yi strcmp ile karşılaştırmak için yazılan pseudo code: strcmp(str1_c[], str2_c[]) olur.

Pseudo koddaki reserved word'ler ve operatörler

- **PROCEDURE MAIN()**: main( ) anlamına gelir
- **BEGIN**: bir bloğun başladığını gösteren süslü parantez ("{") anlamına gelir
- **END**: bir bloğun bittiğini gösteren süslü parantez ("}") anlamına gelir
- **PROCEDURE functionName():** Fonksiyon tanımlarken kullanılan pseudo code syntax'ıdır.
  Örn: func1(x_c, y_ip, z_f) func1 isimli fonksiyon, 3 argüman alır: **char tipinde x parametresi, y isimli bir integer pointer, z isimli bir float.**
- **RETURN VOID:** Fonksiyonun en altında (END'den hemen önce yazılır), değer döndürmeyeceğini belirtir. Herhangi bir RETURN komutu yoksa, fonksiyonun dönüş tipinin void olduğu varsayılır.
  Örn: RETURN a_fp fonksiyon a isimli bir float pointer döndürmektedir.
- **Assignment**: **<-** operatörü (< ve – işaretleri) ile yapılır. Ör: x=5 için pseudo code: x<-5.
- **FOR i_i <- 0..n_i LOOP**: for(i=0; i<n; i++) //NOT: 0..n için: 0'dan n-1'e kadar döndüğü varsayılır
- **WHILE ((i_i<n_i) AND (j_i >m_i)) DO : WHILE ((i<n) && (j>m))**
- **WHILE ((i_i <n_i) OR (j_i >m_i)) DO : WHILE ((i<n) || (j>m))**
- **IF ((i_i <n_i) AND (j_i >m_i)) THEN : IF ((i<n) &&(j>m))**
- **IF ((i_i <n_i) OR (j_i >m_i)) THEN : IF ((i<n) || (j>m))**
- **IF (i_i EQUAL n_i) THEN: if(i == n)**
- **IF (i_i NOT_EQUAL n_i) THEN: if(i != n)**
- **PRINT_LINE "hello world!"** : printf("hello world!\n"); // C koduna çevirirken eklenen newline karakterine dikkat!
- **PRINT_LINE "$i_i + $j_i degeri= $k_i olur"**: printf("%d + %d degeri= %d olur\n", i, j,k); // açıklama **PRINT_LINE** ifadesinde tırnak içinde verilen $ işareti, yanındaki değişkenin (ör: **$i_i** için: integer tipteki "i" değişkeninin) değerini yazmak istediğimiz anlamına gelir. Bu değişken integer ise %d ile, char ise %c ile, … printf() fonksiyonuna aktarılmalıdır.
- **GET_VAL "$arr_i[k_i]"**: arr isimli integer tipteki dizinin k'nci indisteki değerini okuyoruz, yani, scanf("%d", &arr[k]);
- Pseudo koddaki komut satırları newline karakteri ile son bulur, satırların sonunda ";" bulunmaz, C koduna çevirirken, uygun şekilde ";"eklemelisiniz.

NOT: pseudo code'daki reserved word'ler case-sensitive değildir (dolayısıyla örneğin: BEGIN, begin, Begin, … hepsi aynı anlamdadır. Ancak pseudo koddaki değişkenler, C'ye birebir (case-sensitive olarak) aktarılmalıdır (ör: "total_i", C'ye "total" olarak aktarılmalı). C'de değişkenler genelde küçük harf ile yazıldığı için, pseudo kodlarınızda da bu şekilde yazılmalıdırlar.

Örnekler

Aşağıda örnek bir pseudo kod ve bu kodun C karşılığı verilmiştir.

Pseudo code:

```
PROCEDURE MAIN()
BEGIN
     PRINT_LINE "Dizinin eleman sayisini giriniz"
     GET_VAL "$n_i"
     PRINT_LINE "Dizinin elemanlarini giriniz"
       FOR  k_i <- 0..n_i LOOP
       BEGIN
         GET_VAL "$arr_i[k_i]"
        END

     max_i <- 0
     maxind_i <- -1
       FOR  k_i <- 0..n_i LOOP
       BEGIN
              IF (arr_i[k_i] > max_i) THEN
              BEGIN
                   max_i <- arr_i[k_i]
                    maxind_i <- k_i
              END
       END
       PRINT_LINE "Dizinin en buyuk elemani $max_i, $maxind_i. indiste bulunmaktadir."
END
```

C kodu karşılığı:

```
void main(){
   int k, n, max=0, maxind=-1, arr[100];
   printf("Dizinin eleman sayisini giriniz \n");
   scanf("%d", &n);
   printf("Dizinin elemanlarini giriniz \n");
   for (k=0; k<n; k++)   scanf("%d", &arr[k]);

    for (k=0; k<n; k++)   {
      if(arr[k] > max){
         max=arr[k];
         maxind=k;
      }
   }
   printf("Dizinin en buyuk elemani %d, %d. indiste bulunmaktadir.\n", max, maxind);
}
```

Projenin Çalışma Şekli

Okunacak pseudo kodu içeren text dosyasının ismi ve oluşturulacak C kaynak kodunun ismi programınıza komut satırı argumanı olarak aktarılmalıdır. Program doğru parametreler ile çağırılmadığında veya çalışma esnasında bir hata ile karşılaşıldığında gerekli hata mesajları verilerek kullanıcı yönlendirilmelidir.

Aşağıda program için örnek bir çalışma görüntüsü verilmiştir.

```
> PseudocodeConverter pseudoCode1.txt CSource1.c
Pseudocode dosyası okundu
C kaynak dosyası oluşturuluyor
Executable dosya oluşturuldu
```

C kodu içerisinden başka bir C kodunu cc veya gcc gibi derleyiciler ile derleyebilmek için örneğin system() fonksiyonunu kullanabilirsiniz. Ör:

```
int main() {
    ...
    system("gcc -o filename temp.c");
    return 0;
}
```

Ayrıca, C ile oluşturulmuş bir programdan executable dosyaların çalıştırılması için execvp() fonksiyonu kullanılabilir. Oluşturulun C kodunun derlenebilmesi için gcc'nin uygun parametreler ile programınızdan çalıştırılması gerekmektedir. Windows üzerinde gcc ile derleme yapabilmek için Mingw veya CygWin kullanılabilir. Mingw, CygWin, execvp() ve system() ile ilgili detaylı bilgiye aşağıdaki bağlantılardan erişilebilirsiniz.

http://www.mingw.org/
https://www.cygwin.com/
http://man7.org/linux/man-pages/man3/exec.3.html
http://man7.org/linux/man-pages/man3/system.3.html
https://www.geeksforgeeks.org/system-call-in-c/

Projenin Sunulması ve Raporlanması

Projenizi final haftasında daha sonra ilan edilecek bir günde laboratuvar ortamında sunmanız gerekmektedir. Sunulmayan projeler değerlendirmeye alınmayacaktır.

Proje ile birlikte içeriği aşağıda verilmiş bir proje raporunun hazırlanması gerekmektedir:

- Projenin amacını, üretilen çözüm yöntemlerini ve modülleri kısaca anlatan bir özet yazılmalıdır.
- Geliştirilen uygulama için yazılan tüm fonksiyonlar, giriş, çıkış parametreleri ve birbirleri ile olan ilişkilerini de kapsayacak şekilde anlatılmalıdır.

- Kaynak kodlar gerekli açıklamalar ile birlikte eklenmelidir

- Projenin çalışmasını gösteren anlamlı ekran görüntülerine yer verilmelidir.

- Geliştirilen projenin tüm kısıtları ve güçlü yönleri değerlendirilmelidir. Örn: Program nasıl daha iyi hale getirilebilir? C dilinin hangi özelliklerinden yararlanılmıştır?

- Yararlanılan kaynaklar listelenmelidir.