



2018-1

**BLM2031 YAPISAL PROGRAMLAMA
DÖNEM PROJESİ**

04-01-2019

AYŞE HİLAL DOĞAN

17011907

Projenin temel amacı pseudo kodun yazılı olduğu bir text dosyasını, belli kuralları takip ederek bir c kaynak kodu haline getirmek ve bu c kaynak kodunu derleyerek bir exe dosyası oluşturmaktır.

Çözüm yöntemi olarak; text dosyasından her satır başında bir kelime alınıp, bu kelimenin ne olduğuna bağlı olarak, BUL fonksiyonu gidişatı belirler. BUL fonksiyonuna gelen kelime reserved word'lerden hangisi ise, ona göre başka bir fonksiyona gidilir ve bu fonksiyonda fonksiyondan fonksiyona değişmek üzere char char, string string ya da bütün satır alınarak inceleme yapılır ve belli kurallara göre yeni c kaynak kodu adım adım oluşturulur.

```
int main(){  
    FILE *fp,*fp2;  
    char string[100];  
    int c;  
  
    fp=fopen( "pseudoCode1.txt" , "r");  
    if(!fp){  
        fprintf(fp2,"Dosya bulunamadi..\n");  
        return -1;  
    }  
    if( (fp2=fopen( "CSource1.c ", "w"))== NULL){  
        fprintf(fp2,"Dosya acilamadi..\n");  
    }  
    printf("> PseudocodeConverter pseudoCode1.txt CSource1.c\n");  
    printf("Pseudo code dosyasi okundu\n");  
    printf("C kaynak dosyasi olusturuluyor..\n");  
  
    do {  
        c = fscanf(fp,"%s",string); /* dosyadan bir kelime oku... */  
        if (c != EOF){  
            BUL(fp,fp2,string);  
        }  
    } while (c != EOF);          /* EOF olana kadar.. */  
  
    fclose(fp);  
    fclose(fp2);  
    printf("Executable dosya olusturuldu\n");  
    // system("gcc -o filename CSource1.c");  
  
    return 0;  
}
```

Yukarıda verilen main fonksiyonudur. Verilen text dosyasını (pseudoCode1.txt) açar, c kaynak kodunu “w” modunda açar. Gerekli mesajları ekrana verir. pseudoCode1 dosyası EOF olana kadar string string alır ve BUL fonksiyonuna yönlendirir. EOF olunca dosyalar kapatılır ve işlem biter.

```

}
void BUL(FILE *fp ,FILE *fp2, char *string){

    int i=0,j=0,a=0;
    char kelime[100],n[2],tip[100],ch;

    char str[100];
    if( strcmp ( string, "BEGIN" ) ==0 ) {
        fprintf(fp2,"{\n");
    }

    else if( strcmp ( string, "PROCEDURE" ) ==0 ){
        fscanf(fp,"%s",kelime);

        if( strcmp ( kelime, "MAIN" ) ==0 ){
            fprintf(fp2,"void main()\n");
        }

        else{

            fonksiyon_tanimla( kelime ,fp2 );
        }
    }
}

```

```

else if( strcmp ( string, "RETURN" ) ==0 ){
    BILGI tanimla[100];
    int j=0,k=0,i=0;
    fscanf(fp,"%s",kelime);
    while( kelime[i] != '_' ){
        tanimla[0].degisken[j] = kelime[i] ;

        j++;
        i++;
    }
    fprintf(fp2,"return %c;\n",tanimla[0].degisken[0]);
    i++;
    while( i< strlen(kelime)-1){
        tanimla[0].tip[k] = kelime[i] ;
        k++;
        i++;
    }
}

```

```

else if( strcmp ( string, "END" ) ==0 ){
    fprintf(fp2,"}\n");

}

else if( strcmp ( string, "PRINT_LINE" ) ==0 ){
    char ch=34;

    fgets(kelime,100,fp);

    fprintf(fp2,"printf(%c",ch);

    for(i=2 ; i<strlen(kelime)-2 ; i++){
        fprintf(fp2,"%c", kelime[i]);
    }
    fprintf(fp2," %cn%c);\n",92,ch);
}

else if( strcmp ( string, "IF" ) ==0 ){
    fprintf(fp2,"if");
    fgets(kelime,100 , fp);
    while_if(kelime,fp2);
}

```

```

    else if( strcmp ( string, "WHILE" ) ==0 ){
        fprintf(fp2,"while");
        fgets(kelime,100 ,fp);
        while_if(kelime,fp2);
    }
    else if( strcmp ( string, "FOR" ) ==0 ){
        fprintf(fp2,"for(%c",34);

        fgets(kelime,100 ,fp);

        for_f(kelime,fp2);
    }

    else if( strcmp ( string, "GET_VAL" ) ==0 ){
        char ch=34;
        fprintf(fp2,"scanf(");
        fgets(kelime,100 ,fp);
        scan(kelime,fp2);
    }
    else {
        deger_ata(string,fp2);
    }
}

```

Yukarıda görselleri verilen **BUL** fonksiyonu mainden sonra gidilen ilk fonksiyondur.

Mainden gelen kelime BEGIN ise kaynak koduna '{' yazar, PROCEDURE ise iki ihtimal vardır(text dosyasından yen bir kelime alınır); ya maindir ya fonksiyon tanımlamasıdır. Buna göre mainse void main() yazılır, fonksiyon ise **fonksiyon_tanimla** fonksiyonuna gidilir.İşlemler orada devam eder.

Eğer gelen kelime RETURN ise gelen format şuna benzer olur : RETURN a_fp . '_' ye kadar tutulan structure'ın degisken field'ına kaydedilir, '_' den stringin bitimine kadar ise tip field'ına kaydedilir. Kaynak dosyasına return (degisken ismi) yazılır.

Eğer gelen kelime END ise c kaynak koduna '}' yazılır.

Eğer gelen kelime PRINT_LINE ise il önce printf(" yazılır daha sonra mevcut satır alınır ve kelimenin yazılmak istenen kısmı char char yazılır, ve "); yazılarak çıkılır.

Eğer gelen kelime IF ise ya da while ise mevcut satır alınır ve **while_if** fonksiyonuna gidilir.

Eğer gelen kelime FOR ise mevcut satır alınır ve **for_f** fonksiyonuna gidilir.

Eğer gelen kelime GET_VAL ise scanf(" yazılır ve **scan** fonksiyonuna gidilir.

Eğer bunlardan hiçbiri değilse değer atamadır(x < 5) ve **deger_ata** fonksiyonuna gidilir.

```
void tip_yaz(BILGI *tanimla , FILE *fp2,int a){
    int i;

    for(i=0 ; i<a ; i++){

        if( strcmp( tanimla[i].tip , "c" ) == 0 ){
            fprintf(fp2,"char %s ", tanimla[i].degisken);
        }
        else if( strcmp( tanimla[i].tip , "f" ) ==0 ){
            fprintf(fp2,"float %s ", tanimla[i].degisken);
        }
        else if( strcmp( tanimla[i].tip , "i" ) ==0 ){
            fprintf(fp2,"int %s ", tanimla[i].degisken);
        }
        else if( strcmp( tanimla[i].tip , "l" ) ==0 ){
            fprintf(fp2,"long %s ", tanimla[i].degisken);
        }
        else if( strcmp( tanimla[i].tip , "d" ) ==0 ){
            fprintf(fp2,"double %s ", tanimla[i].degisken);
        }
        else if( strcmp( tanimla[i].tip , "ld" ) ==0 ){
            fprintf(fp2,"long double %s ", tanimla[i].degisken);
        }
    }
}
```

```

else if( strcmp( tanimla[i].tip , "ld" ) ==0 ){
    fprintf(fp2,"long double %s ", tanimla[i].degisken);
}
else if( strcmp( tanimla[i].tip , "file" ) ==0 ){
    fprintf(fp2,"FILE * %s ", tanimla[i].degisken);
}
else if( strcmp( tanimla[i].tip , "cp" ) ==0 ){
    fprintf(fp2,"char * %s ", tanimla[i].degisken);
}
else if( strcmp( tanimla[i].tip , "fp" ) ==0 ){
    fprintf(fp2,"float * %s ", tanimla[i].degisken);
}
else if( strcmp( tanimla[i].tip , "ip" ) ==0 ){
    fprintf(fp2,"int * %s ", tanimla[i].degisken);
}
else if( strcmp( tanimla[i].tip , "lp" ) ==0 ){
    fprintf(fp2,"long * %s ", tanimla[i].degisken);
}
else if( strcmp( tanimla[i].tip , "dp" ) ==0 ){
    fprintf(fp2,"double * %s ", tanimla[i].degisken);
}
else if( strcmp( tanimla[i].tip , "ldp" ) ==0 ){
    fprintf(fp2,"long double * %s ", tanimla[i].degisken);
}
if(i<a-1)
    fprintf(fp2,",");
}
fprintf(fp2,");\n");
}

```

Yukarıda verilen **tip_yaz** fonksiyonu **fonksiyon_tanimla** fonksiyonunda degisken ve tip isimleri kaydedildikten sonra bunları fonksiyon tanımlaması için yazan fonksiyondur.

Sıra sıra if'lerle tip'in ne olduğuna bağlı olarak char/int/long gibi tanımlamalar ve degisken ismini yazar,
 (tanimla.degisken = " x " , tanimla.tip = " ip ") -> (" float * x ")

```

void fonksiyon_tanimla( char *string ,FILE *fp2){
    BILGI tanimla[100];
    int i=0,j=0,k=0;
    int a=0;
    fprintf(fp2,"void ");
    while( string[i] != '\n'){
        fprintf(fp2,"%c", string[i]);
        i++;
    }
    fprintf(fp2,"(");
    i++;
    while( i < strlen(string)){
        j=0,k=0;
        while( string[i] != ' ' ){
            tanimla[a].degisken[j] = string[i] ;
            j++;
            i++;
        }
        i++;
        while( string[i] != ',' && i<strlen(string)){
            if(string[i] == ' '){
                i++;
            }
            else{
                tanimla[a].tip[k] = string[i] ;
                k++;
                i++;
            }
        }
        i++;
        a++;
    }
    tip_yaz(tanimla,fp2,a);
}

```

Yukarıda verilen **fonksiyon_tanimla** fonksiyonu ; PROCEDURE'den sonra main değilse gelen fonksiyondur. Fonksiyonun ismini belirler ve c kaynak koduna yazar, daha sonra degisken tip ve isimlerini structure'a kaydeder ve kaydettiği verileri **tip_yaz** fonksiyonuna yollar ve orada gereken tanımlamaları yapar.

```

} void deger_ata(char *string,FILE *fp2){
    int i=0;
    while( string[i] != '<'){
        fprintf(fp2,"%c", string[i]);
        i++;
    }
    fprintf(fp2,"=");
    i+=2;
    while(i<strlen(string)){
        fprintf(fp2,"%c ",string[i]);
        i++;
    }
    fprintf(fp2,";\n");
}

```

Yukarıda verilen **deger_ata** fonksiyonu ($x \leftarrow 5$) gibi bir tanımlamayı $x=5$; şekline çevirir.

```
void while_if(char *line, FILE *fp2){
    BILGI tanimla[100];
    int i=0,j,k=0;
    int a,b=0;
    char ch;
    while( i< strlen(line) ){
        if( (line[i] == '(' ) ){
            fprintf(fp2,"(");
            i++;
            if( line[i] != ' '){
                for(a=0; a<2 ; a++){
                    j=0;
                    k=0;
                    while( line[i] != ' '){
                        tanimla[a].degisken[j] = line[i] ;
                        fprintf(fp2,"%c",tanimla[a].degisken[j] );
                        j++;
                        i++;
                    }
                }
                i++;
            }
        }
    }
}
```

```
        while( (line[i] != '<' ) && (line[i] != '>' ) && (line[i] != ' ' ) && (line[i] != 'E' ) && (line[i] != 'N' ) ){
            tanimla[a].tip[k] = line[i] ;
            k++;
            i++;
        }
        if((line[i] != 'E') && (line[i] != 'N')){
            fprintf(fp2,"%c",line[i]);
            i++;
        }
        else if(line[i] == 'E' ){
            fprintf(fp2,"==");
            i+=6;
        }
        else if(line[i] == 'N' ){
            fprintf(fp2,"!=");
            i+=10;
        }
    }
}
else if( (line[i] == 'T' ) ){
    fprintf(fp2,")\n");
    i= (strlen(line)+1);
}
}
```

```
    else if( (line[i] == 'D' ) ){
        fprintf(fp2,")\n");
        i= (strlen(line)+1);
    }
    else if( (line[i] == 'A' ) ){
        i+=3;
        fprintf(fp2,"&&");
    }
    else if( (line[i] == 'O' ) ){
        i+=2;
        fprintf(fp2,"||");
    }
    else if( line[i] == ' '){
        i++;
    }
    else i++;
}
}
```


Yukarıda verilen **while_if** fonksiyonu BUL fonksiyonunda gelen kelime WHILE ya da IF ise gelinen fonksiyondur. Mevcutta BUL fonksiyonundan gelen mevcut satır vardır. ' i ' parametresi her işlem yapıldığında işleme göre artış gösterir ve strlen(line)'ın sonuna gelindiğinde işlem biter.

Örnek while: WHILE ((i_i EQUAL n_i) OR (k_f>m_i)) DO

Parantezler control edilerek başlanır. ' _ ' görene kadar struct'ın degisken kısmına kayıt yapar ve c kaynak koduna yazar. EQUAL,>,<,NOT EQUAL vb bir şey görene kadar da tip kısmına kayıt yapar. Tip tanımlaması yaptıktan sonra karşılaştığı char a göre (E ise EQUAL demektir (==) , < ise '<') kaynak koduna yazar.

```
void for_f(char *line, FILE *fp2){
    BILGI tanimla[100];
    int i=0,j,k=0;
    int a,b=0;
    char ch;
    while( i< strlen(line) ){
        for(a=0;a<2;a++){
            j=0;
            k=0;
            while( line[i] != '_' ){
                tanimla[a].degisken[j] = line[i] ;
                j++;
                i++;
            }
            i++;
            while( line[i] != '<' && line[i] != 'L'){
                tanimla[a].tip[k] = line[i] ;
                k++;
                i++;
            }
            if(line[i] == '<'){
                ch=line[i];
                i+=3;
                fprintf(fp2,"%s=%c;",tanimla[a].degisken , line[i] );
                i+=2;
            }
            else if(line[i]=='L'){
                fprintf(fp2,"%s%c%s;%s++%c)\n",tanimla[0].degisken,ch,tanimla[1].degisken,tanimla[0].degisken,34);
                i=strlen(line);
            }
            i++;
        }
    }
}
```

Yukarıda verilen **for_f** fonksiyonu BUL fonksiyonunda gelen kelime FOR ise gelinen fonksiyondur.

Örnek for yapısı: FOR i_i <- 0..n_i LOOP

' _ ' görene kadar structure'ın degisken kısmına kaydeder. ' _ ' gördükten sonra atama işareti olan '<' görene kadar tip kısmına kaydeder. Atanan sayıyı ve degiskeni (i=0) şeklinde kaynak koduna yazar. Eğer 'L' (LOOP 'un L' si) görülmediyse döngünün başına döner ve diğer degiskenin kaydını aynı şekilde yapar. 'L' görünce gerekli yazdırmaları yapar ve iş bitirilir.

Dönüşen for yapısı : for(" i=0; i<n; i++")

```

void scan(char *line , FILE *fp2 ){
    BILGI tanimla;
    char ch=34;
    int i=2,k=0,j=0;

    while( i< strlen(line)-2 ){
        if(line[i] == '$'){
            i++;
            while( line[i] != '_' ){
                tanimla.degisken[j] = line[i] ;
                j++;
                i++;
            }
            i++;
            while( i< strlen(line)-2 ){
                tanimla.tip[k] = line[i] ;
                k++;
                i++;
            }
        }
        i++;
    }
    fprintf(fp2, " %c%c%c%c,%c\n",ch,37,tanimla.tip[0],ch, tanimla.degisken[0]);
}

```

Yukarıda verilen **scan** fonksiyonu BUL fonksiyonunda GET_VAL görüldüğünde gelinen fonksiyondur.

Örnek scanf pseudo kodu : GET_VAL "\$n_i"

'\$' görünce '_' olana kadar tanimla struct'ının degisken kısmına degisken ismini kaydeder, '_' den sonrasını tip kısmına kaydeder.

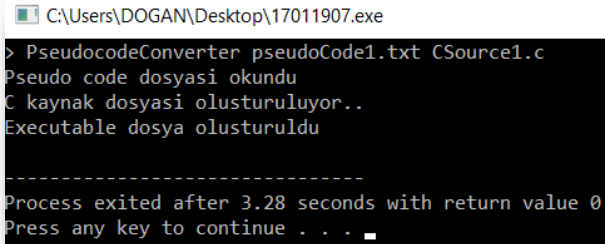
Kayıtlar yapıldıktan sonra (scanf("%i",&n);) bu formatta kaynak koduna yazılır.

***Program nasıl daha iyi hale getirilebilir? C dilinin hangi özelliklerinden yararlanılmıştır?**

Program henüz tam değildir. Degisken tanımlamaları yapılmamıştır. Fonksiyon geri dönüş tipi güncellenmemiştir.

Program bunlar yapılarak daha iyi bir hale getirilebilir. C dilinin structure özelliğinden, dosyayı herhangi bir modda açma özelliğinden ve bütün diğer özelliklerden faydalanılmıştır.

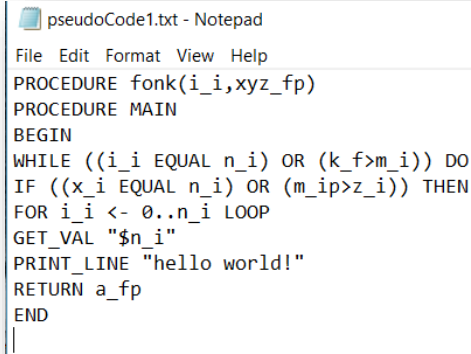
*Ekran görüntüleri:



```
C:\Users\DOGAN\Desktop\17011907.exe
> PseudocodeConverter pseudoCode1.txt CSource1.c
Pseudo code dosyasi okundu
C kaynak dosyasi olusturuluyor..
Executable dosya olusturuldu

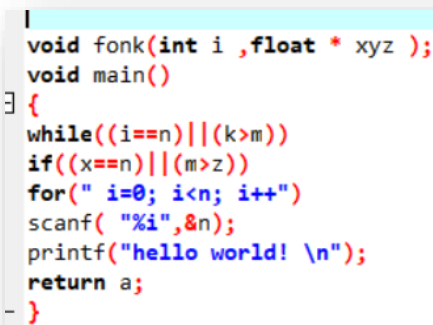
-----
Process exited after 3.28 seconds with return value 0
Press any key to continue . . .
```

Text dosyası:



```
pseudoCode1.txt - Notepad
File Edit Format View Help
PROCEDURE fonk(i_i,xyz_fp)
PROCEDURE MAIN
BEGIN
WHILE ((i_i EQUAL n_i) OR (k_f>m_i)) DO
IF ((x_i EQUAL n_i) OR (m_ip>z_i)) THEN
FOR i_i <- 0..n_i LOOP
GET_VAL "$n_i"
PRINT_LINE "hello world!"
RETURN a_fp
END
|
```

Dönüştürülen c kaynak kodu:



```
1
void fonk(int i ,float * xyz );
void main()
{
while((i==n)||((k>m))
if((x==n)||((m>z))
for(" i=0; i<n; i++")
scanf( "%i",&n);
printf("hello world! \n");
return a;
- }
```

*Yararlanılan kaynaklar :

<http://www.asciitable.com>

www.codingunit.com