

buffer.c File Reference

Dairesel (circular) karakter tamponu (FIFO) uygulaması. [More...](#)

```
#include <stdio.h>
#include <string.h>
```

Go to the source code of this file.

Macros

```
#define BUYUKLUK 10
    Tamponun (buffer) maksimum kapasitesi.
```

Functions

```
int push (char value)
    Buffer'a karakter ekler (push işlemi)

int pop (char *value)
    Buffer'dan karakter okur (pop işlemi)

void buffer_goster ()
    Buffer'in mevcut durumunu grafiksel olarak ekrana yazdırır.

int main ()
    Programın giriş noktası (main fonksiyonu)
```

Variables

```
char buffer [BUYUKLUK]
    Dairesel tamponu temsil eden global değişkenler.

int head = 0
int tail = 0
int count = 0
```

Detailed Description

Dairesel (circular) karakter tamponu (FIFO) uygulaması.

Bu program, sabit boyutlu bir tampon (buffer) kullanarak karakter ekleme ve çıkarma işlemleri yapar. Tampon bir kuyruk (queue) mantığı ile çalışır: İlk giren karakter ilk çıkar (FIFO).

Kullanıcıya bir menu sunulur. Bu menu aracılığıyla kullanıcı:

- Çoklu karakter girişi yaparak buffer'a veri ekleyebilir,
- Mevcut buffer'dan bir karakter okuyabilir (çıkartabilir),
- Tamponun mevcut durumunu görebilir,
- Veya çıkış yapabilir.

Definition in file `buffer.c`.

Macro Definition Documentation

BUYUKLUK

```
#define BUYUKLUK 10
```

Tamponun (buffer) maksimum kapasitesi.

< Giriş/Çıkış işlemleri için gerekli kutuphane < Karakter dizisi işlemleri için kutuphane

Definition at line 22 of file `buffer.c`.

Function Documentation

buffer_goster()

```
void buffer_goster ( )
```

Buffer'in mevcut durumunu grafiksel olarak ekrana yazdırır.

Dolular karakterlerle, boş alanlar '-' ile gösterilir.

Definition at line 73 of file `buffer.c`.

```
73 {
74     printf("Buffer durumu: [ ];\n");
75     for (int i = 0; i < BUYUKLUK; i++) {
76         if (count == 0) {
77             printf(" ");
78         } else {
79             int aktif = 0;
80             if (tail < head) {
81                 if (i >= tail && i < head) aktif = 1;
82             } else if (tail > head) {
83                 if (i >= tail || i < head) aktif = 1;
84             } else if (count == BUYUKLUK) {
85                 aktif = 1;
86             }
87             if (aktif) {
88                 printf(" %c", buffer[i]);
89             } else {
90                 printf(" ");
91             }
92         }
93     }
94     printf("\n");
95 }
96 }
```

◆ main()

int main ()

Programın giriş noktası (main fonksiyonu)

Kullanıcıya etkileşimli bir menü sunar.

1. Çoklu karakter ekleme
2. FIFO'dan karakter okuma

1. Programdan çıkış

Returns

int Programdan çıkış kodu

Definition at line 108 of file `buffer.c`.

```
108 |     {
109 |         int secim;
110 |         char giris[100];
111 |         char alinan;
112 |
113 |         while (1) {
114 |             printf("\n1 - Ekle (çoklu harf veya rakam)\n2 - Oku (cikart)\n0 - Cikis\nSecim: ");
115 |             scanf("%d", &secim);
116 |             while ((getchar()) != '\n'); // Enter karakterini temizle
117 |
118 |             switch (secim) {
119 |                 case 1:
120 |                     printf("Giris (birden fazla harf/rakam yaz): ");
121 |                     fgets(giris, sizeof(giris), stdin);
122 |                     giris[strcspn(giris, "\n")] = '\0'; // Satir sonunu kaldır
123 |
124 |                     for (int i = 0; i < strlen(giris); i++) {
125 |                         if (push(giris[i])) {
126 |                             printf("Eklendi: %c\n", giris[i]);
127 |                         } else {
128 |                             printf("Buffer dolu, eklenemedi: %c\n", giris[i]);
129 |                             break;
130 |                         }
131 |                     }
132 |                     buffer_goster();
```

◆ pop()

int pop (char *value)

Buffer'dan karakter okur (pop işlemi)

Eğer buffer bossa karakter okunamaz.

Parameters

value Okunan karakterin yazılacağı adres

Returns

int 1 başarılı, 0 başarısız (buffer boş)

Definition at line 58 of file `buffer.c`.

```
58 |     {
59 |         if (count == 0) {
60 |             return 0; // Bos
61 |         }
62 |         *value = buffer[tail];
63 |         tail = (tail + 1) % BUYUKLUK;
64 |         count--;
65 |         return 1;
66 |     }
```