buffer.c File Reference

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

Go to the source code of this file.

Macros

#define	BUYUKLUK 10

Functions

int	push (char value)
int	pop (char *value)
void	buffer_goster ()
int	main ()

Variables

char	buffer [BUYUKLUK]
int	head = 0
int	tail = 0
int	count = 0

Macro Definition Documentation

◆ BUYUKLUK

```
#define BUYUKLUK   10
```

Definition at line **4** of file **buffer.c**.

Function Documentation

◆ buffer_goster()

```
void buffer_goster ( )
```

Definition at line **31** of file **buffer.c**.

```
31                          {
32      printf("Buffer durumu: [");
33      for (int i = 0; i < BUYUKLUK; i++) {
34          if (count == 0) {
35              printf(" -");
36          } else {
37              int aktif = 0;
38              if (tail < head) {
39                  if (i >= tail && i < head) aktif = 1;
40              } else if (tail > head) {
41                  if (i >= tail || i < head) aktif = 1;
42              } else if (count == BUYUKLUK) {
43                  aktif = 1;
44              }
45
46              if (aktif) {
47                  printf(" %c", buffer[i]);
48              } else {
49                  printf(" -");
50              }
51          }
52      }
53      printf(" ]\n");
54  }
```

◆ main()

```
int main ( )
```

Definition at line **56** of file **buffer.c**.

```
 56             {
 57      int secim;
 58      char giris[100];
 59      char alinan;
 60  
 61      while (1) {
 62          printf("\n1 - Ekle (coklu harf veya rakam)\n2 - Oku (cikart)\n0 - Cikis\nSecim: ");
 63          scanf("%d", &secim);
 64          while ((getchar()) != '\n');
 65  
 66          switch (secim) {
 67              case 1:
 68                  printf("Giris (birden fazla harf/rakam yaz): ");
 69                  fgets(giris, sizeof(giris), stdin);
 70                  giris[strcspn(giris, "\n")] = '\0';
 71  
 72                  for (int i = 0; i < strlen(giris); i++) {
 73                      if (push(giris[i])) {
 74                          printf("Eklendi: %c\n", giris[i]);
 75                      } else {
 76                          printf("Buffer dolu, eklenemedi: %c\n", giris[i]);
 77                          break;
 78                      }
 79                  }
 80                  buffer_goster();
 81                  break;
 82  
 83              case 2:
 84                  if (pop(&alinan)) {
 85                      printf("Okunan karakter: %c\n", alinan);
 86                  } else {
 87                      printf("Buffer bos!\n");
 88                  }
 89                  buffer_goster();
 90                  break;
 91  
 92              case 0:
 93                  printf("Programdan cikiliyor...\n");
 94                  return 0;
 95  
 96              default:
 97                  printf("Gecersiz secim!\n");
 98          }
 99      }
100  
101      return 0;
102  }
```

◆ pop()

```
int pop ( char * value )
```

Definition at line **21** of file **buffer.c**.

```
21                      {
22      if (count == 0) {
23          return 0; // Bo�
24      }
25      *value = buffer[tail];
26      tail = (tail + 1) % BUYUKLUK;
27      count--;
28      return 1;
29  }
```

◆ push()

```
int push ( char value )
```

Definition at line **11** of file **buffer.c**.

```
11                      {
12      if (count == BUYUKLUK) {
13          return 0; // Dolu
14      }
15      buffer[head] = value;
16      head = (head + 1) % BUYUKLUK;
17      count++;
18      return 1;
19  }
```

Variable Documentation

◆ buffer

```
char buffer[BUYUKLUK]
```

Definition at line **6** of file **buffer.c**.

◆ count

```
int count = 0
```

Definition at line **9** of file **buffer.c**.

◆ head

```
int head = 0
```

Definition at line **7** of file **buffer.c**.

◆ tail

```
int tail = 0
```

Definition at line **8** of file **buffer.c**.

Generated by doxygen 1.14.0