Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №5
по дисциплине «Операционные системы»

Тема Буферизованный ввод / вывод

Студент Романов А.В.

Группа ИУ7-63Б

Оценка (баллы) _____

Преподаватель Рязанова Н. Ю.

Москва — 2021 г.

1 Первая программа

Листинг 1.1: Программа №1

```c
#include <stdio.h>
#include <fcntl.h>

#define OK 0
#define BUF_SIZE 20
#define VALID_READED 1

#define FILE_NAME "data/alphabet.txt"
#define SPEC "%c"

int main(void)
{
  int fd = open(FILE_NAME, O_RDONLY);

  FILE *fs1 = fdopen(fd, "r");
  char buff1[BUF_SIZE];
  setvbuf(fs1, buff1, _IOFBF, BUF_SIZE);

  FILE *fs2 = fdopen(fd, "r");
  char buff2[BUF_SIZE];
  setvbuf(fs2, buff2, _IOFBF, BUF_SIZE);

  int flag1 = 1, flag2 = 2;
  while (flag1 == VALID_READED || flag2 == VALID_READED)
  {
    char c;

    if ((flag1 = fscanf(fs2, SPEC, &c)) == VALID_READED)
    {
      fprintf(stdout, SPEC, c);
    }

    if ((flag2 = fscanf(fs2, SPEC, &c) == VALID_READED)
    {
      fprintf(stdout, SPEC, c);
    }
  }
```

```
38
39    return OK;
40 }
```

Листинг 1.2: Программа №1 (реализация с потоками)

```
1  #include <stdio.h>
2  #include <fcntl.h>
3  #include <pthread.h>
4
5  #define OK 0
6  #define BUF_SIZE 20
7  #define VALID_READED 1
8
9  #define FILE_NAME "data/alphabet.txt"
10 #define SPEC "%c"
11
12 void *run_buffer(void *args)
13 {
14   int flag = 1;
15   FILE *fs = (FILE *)args;
16
17   while (flag == VALID_READED)
18   {
19     char c;
20     if ((flag = fscanf(fs, SPEC, &c)) == VALID_READED)
21     {
22       fprintf(stdout, SPEC, c);
23     }
24   }
25
26   return NULL;
27 }
28
29 int main(void)
30 {
31   setbuf(stdout, NULL);
32   pthread_t thread;
33   int fd = open(FILE_NAME, O_RDONLY);
34
35   FILE *fs1 = fdopen(fd, "r");
36   char buff1[BUF_SIZE];
37   setvbuf(fs1, buff1, _IOFBF, BUF_SIZE);
38
39   FILE *fs2 = fdopen(fd, "r");
40   char buff2[BUF_SIZE];
41   setvbuf(fs2, buff2, _IOFBF, BUF_SIZE);
42
43   int rc = pthread_create(&thread, NULL, run_buffer, (void *)fs2);
```

```
44
45    int flag = 1;
46    while (flag == VALID_READED)
47    {
48      char c;
49      fprintf(stdout, "\nSCANF IN MAIN_1");
50      flag = fscanf(fs1, SPEC, &c);
51      fprintf(stdout, "\nSCANF IN MAIN_2");
52      if (flag == 1)
53      {
54        fprintf(stdout, SPEC, c);
55      }
56    }
57
58    pthread_join(thread, NULL);
59    return OK;
60  }
```

2 | Вторая программа

Листинг 2.1: Программа №2

```c
#include <fcntl.h>
#include <unistd.h>

#define OK 0
#define VALID_READED 1
#define FILE_NAME "data/alphabet.txt"

int main(void)
{
  int fd1 = open(FILE_NAME, O_RDONLY);
  int fd2 = open(FILE_NAME, O_RDONLY);
  int rc1, rc2 = VALID_READED;

  while (rc1 == VALID_READED || rc2 == VALID_READED)
  {
    char c;

    rc1 = read(fd1, &c, 1);
    if (rc1 == VALID_READED)
    {
      write(1, &c, 1);
    }

    rc2 = read(fd2, &c, 1);
    if (rc2 == VALID_READED)
    {
      write(1, &c, 1);
    }
  }

  return OK;
}
```

Листинг 2.2: Программа №2 (реализация с потоками)

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

```c
#include <fcntl.h>
#include <unistd.h>
#include <pthread.h>

#define OK 0
#define VALID_READED 1
#define FILE_NAME "data/alphabet.txt"

void *run_buffer(void *args)
{
  int fd = *((int *)args);
  int err = VALID_READED;

  while (err == VALID_READED)
  {
    char c;
    err = read(fd, &c, 1);
    if (err == VALID_READED)
    {
      write(1, &c, 1);
    }
  }

  return NULL;
}

int main(void)
{
  int fd1 = open(FILE_NAME, O_RDONLY);
  int fd2 = open(FILE_NAME, O_RDONLY);

  pthread_t thread;
  int rc = pthread_create(&thread, NULL, run_buffer, (void *)(&fd2));
  int err = VALID_READED;

  while (err == VALID_READED)
  {
    char c;
    err = read(fd1, &c, 1);
    if (err == VALID_READED)
    {
      write(1, &c, 1);
    }
  }

  pthread_join(thread, NULL);
  return OK;
}
```

3 | Третья программа

Листинг 3.1: Программа №2

```c
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>

#define OK 0
#define FILE_NAME "data/out.txt"
#define SPEC "%c"

int main()
{
  FILE *f1 = fopen(FILE_NAME, "w");
  FILE *f2 = fopen(FILE_NAME, "w");

  for (char c = 'a'; c <= 'z'; c++)
  {
    if (c % 2)
    {
      fprintf(f1, SPEC, c);
    }
    else
    {
      fprintf(f2, SPEC, c);
    }
  }

  fclose(f2);
  fclose(f1);

  return OK;
}
```

Листинг 3.2: Программа №3 (реализация с потоками)

```c
#include <stdio.h>
#include <fcntl.h>
#include <pthread.h>
```

```c
#include <unistd.h>

#define OK 0
#define FILE_NAME "data/out.txt"
#define SPEC "%c"

void *run_buffer(void *args)
{
  FILE *f = (FILE *)args;

  for (char c = 'b'; c <= 'z'; c += 2)
  {
    fprintf(f, SPEC, c);
  }

  fclose(f);
  return NULL;
}

int main()
{
  FILE *f1 = fopen(FILE_NAME, "w");
  FILE *f2 = fopen(FILE_NAME, "w");

  pthread_t thread;
  int rc = pthread_create(&thread, NULL, run_buffer, (void *)(f2));

  for (char c = 'a'; c <= 'z'; c += 2)
  {
    fprintf(f1, SPEC, c);
  }

  pthread_join(thread, NULL);
  fclose(f1);

  return OK;
}
```