

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №5
по дисциплине: **ОСИСП**
Тема: Ввод/вывод

Выполнил
студент 2 курса
Корнаसेвич И. Д.

Проверил
Давидюк Ю. И.

Задание Написать программу, которая получает со стандартного потока ввода список файлов каталога, и выводит их в стандартный поток вывода, добавляя перед каждым именем порядковый номер. Протестировать на различных каталогах с использованием конвейеров в различных комбинациях вашей программы и команд `ls`, `sort`, `head`, `tail`. В текущем каталоге создайте два файла, в один из которых выводите нечетные, а во второй — четные порядковые номера и соответствующие имена файлов.

main.c

```
1  #include <stdio.h>
2  #include <unistd.h>
3  #include <string.h>
4  #include <uv.h>
5
6  ssize_t writeToEnd(int fd, char* buf, int bufSize){
7      int l = 0;
8      int r = 0;
9      while (buf[r] != '\0' && r < bufSize){
10         r++;
11     }
12     return write(fd, buf, r - l);
13 }
14
15 int main() {
16     int bufSize = 255;
17     int numSize = 10;
18     char buf[bufSize];
19     char num[numSize];
20     bzero(num, numSize);
21     bzero(buf, bufSize);
22     int fd = 1;
23     (void)umask(0);
24     int fdOdd = open("odd", O_WRONLY | O_CREAT, 0666);
25     int fdEven = open("even", O_WRONLY | O_CREAT, 0666);
26     if (fdOdd < 0 || fdEven < 0){
27         printf("Cannot open a file.");
28         return -1;
29     }
30
31     for (int i = 0; read(0, buf, bufSize) != 0; bzero(buf, bufSize)){
32         int l = 0;
33         int r = 0;
34         while (1){
35             while (buf[r] != '\n' && buf[r] != '\0' && r < bufSize){
36                 r++;
37             }
38
39             if (buf[r] == '\0' || r == bufSize){
40                 break;
41             }
42             sprintf(num, "%d. ", i);
43
44             if (i % 2 == 0){
45                 fd = fdEven;
46             } else {
```

```

47         fd = fdOdd;
48     }
49
50     writeToEnd(fd, num, numSize);
51     write(fd, &buf[l], r - l + 1);
52     i++;
53     l = ++r;
54 }
55 }
56 if (close(fdOdd) < 0 || close(fdEven) < 0){
57     printf("Cannot close a file.");
58     return -1;
59 }
60 }

```

Результаты работы при вызове: `ls | ./src`

odd.

```

1 1. CMakeFiles
2 3. even
3 5. myfile
4 7. src
5 9. Testing

```

even.

```

1 0. CMakeCache.txt
2 2. cmake_install.cmake
3 4. Makefile
4 6. odd
5 8. src.cbp

```

Результаты работы при вызове: `ls | head -n 3 | ./src`

oddHead.

```

1 1. CMakeFiles

```

evenHead.

```

1 0. CMakeCache.txt
2 2. cmake_install.cmake

```

Вывод: Я познакомился с файловыми дескрипторами, системными вызовами `open()`, `write()`, `read()`, `close()`. Использовал `isatty()` для считывания `stdin`, и записи данных в файл.