

Лабораторная работа 6.

РАБОТА С КАНАЛАМИ (Pipes)

Цель. Освоить механизм взаимодействия между процессами на основе каналов (pipes).

Задания

Задание1. Получите через канал системное время и дату.

```
1  #include <iostream>
2  #include <stdlib.h>
3  #include <stdio.h>
4  #include <string.h>
5
6  using namespace std;
7
8  int main()
9  {
10     FILE *read_fp;
11     char buffer[BUFSIZ + 1];
12     int chars_read;
13
14     memset(buffer, '\0', sizeof(buffer));
15
16     read_fp = popen("date", "r");
17
18     if (read_fp != NULL)
19     {
20         chars_read = fread(buffer, sizeof(char), BUFSIZ, read_fp);
21         if (chars_read > 0)
22         {
23             printf("\n%s\n", buffer);
24         }
25         pclose(read_fp);
26         exit(EXIT_SUCCESS);
27     }
28     exit(EXIT_FAILURE);
29 }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
gost@samurai:~/University/3_1/LinuxLabs/lab01$ g++ date.cpp
gost@samurai:~/University/3_1/LinuxLabs/lab01$ ./a.out

Tue 28 Sep 2021 06:51:18 AM +03
```

Задание 2. Получите через канал содержимое текстового файла.

```
1  #include <iostream>
2  #include <stdlib.h>
3  #include <stdio.h>
4  #include <string.h>
5
6  using namespace std;
7
8  int main()
9  {
10     FILE *read_fp;
11     char buffer[BUFSIZ + 1];
12     int chars_read;
13
14     memset(buffer, '\0', sizeof(buffer));
15
16     read_fp = popen("cat txt.txt", "r");
17
18     if (read_fp != NULL)
19     {
20         chars_read = fread(buffer, sizeof(char), BUFSIZ, read_fp);
21         if (chars_read > 0)
22         {
23             printf("\n%s\n", buffer);
24         }
25         pclose(read_fp);
26         exit(EXIT_SUCCESS);
27     }
28     exit(EXIT_FAILURE);
29 }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
gost@samurai:~/University/3_1/LinuxLabs/lab01$ g++ cat.cpp
gost@samurai:~/University/3_1/LinuxLabs/lab01$ ./a.out

This is the test file
With love. Linux community <3
gost@samurai:~/University/3_1/LinuxLabs/lab01$
```

Задание. Предыдущий пример исправьте так, чтобы в выводе на консоль заменить слово Linux на Laba.

```
1  #include <unistd.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <stdio.h>
5  #define BUFSIZZ 256
6
7  int main()
8  {
9      FILE* write_fp;
10     char buffer[BUFSIZZ+1];
11     sprintf(buffer, "Welcome to Linux community!\n");
12
13     write_fp = popen("cat", "w");
14
15     fwrite(buffer, sizeof(char), strlen(buffer), write_fp);
16
17     pclose(write_fp);
18
19     exit(0);
20
21 }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
gost@samurai:~/University/3_1/LinuxLabs/lab01$ g++ write.cpp
gost@samurai:~/University/3_1/LinuxLabs/lab01$ ./a.out
Welcome to Linux community!
gost@samurai:~/University/3_1/LinuxLabs/lab01$ █
```