

Лекция 13

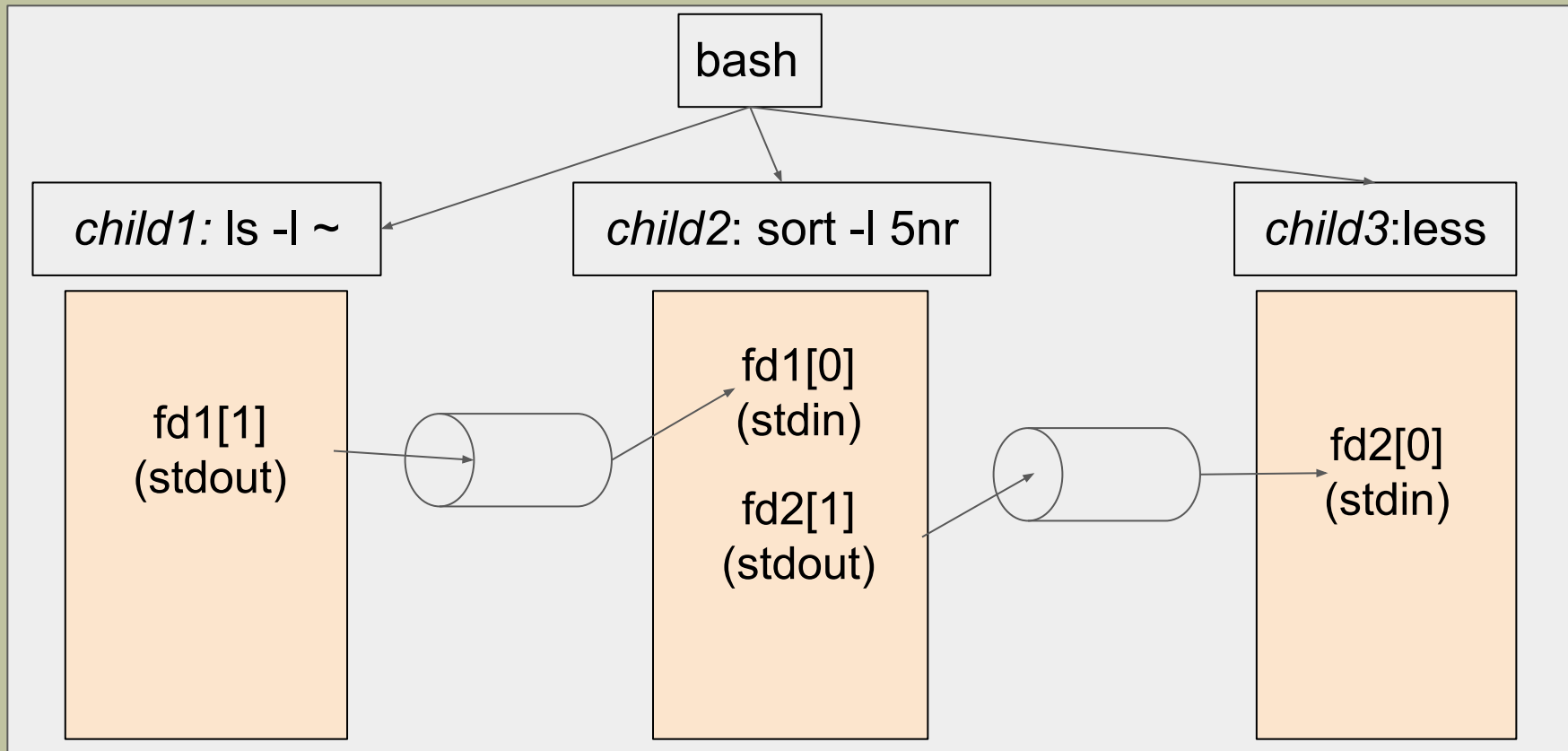
- Неименованные каналы.
- Каналы FIFO (именованные каналы).
- Именованные каналы в MS Windows

Неименованные каналы

```
> ls -l ~ | sort -k 5nr | less
```

```
-rwxr-xr-x  1 malkov users 3532172082 Oct 17  2020  
cuda_11.1.1_455.32.00_linux.run  
-rwxr-xr-x  1 malkov users 3066694836 Jul  3  2020  
cuda_11.0.2_450.51.05_linux.run  
-rw-r--r--  1 malkov users  10432532 Jun 13  2021  
Image-0001.png  
-rw-r--r--  1 malkov users  194745 Sep 17 14:42  
oscSL-W.png  
-rw-r--r--  1 malkov users  183002 Sep 17 14:44  
oscSL-U.png
```

Lines 2-6



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

```
int main(){
```

```
    fprintf(stdout, "sincerely yours, ");
```

```
    return 0;
```

```
}
```

lab13Pa-1.c

```
#include <stdio.h>
#include <string.h>
#define BUFFER_SIZE 128
```

lab13Pa-2.c

```
int main(){
    char buf[BUFFER_SIZE];
    memset(buf, 0, BUFFER_SIZE);
    fgets(buf, BUFFER_SIZE, stdin);
```

```
    buf[0]-=0x20;
    strcat(buf, "E.A. Malkov\n");
```

```
    fprintf(stdout, "%s", buf);
    return 0;
}
```

```
> ./lab13Pa-1  
sincerely yours,
```

```
> ./lab13Pa-2  
q  
Q  
E.A. Malkov
```

```
> ./lab13Pa-1 | ./lab13Pa-2  
Sincerely yours, E.A. Malkov
```

```
> ./lab13Pa  
Sincerely yours, E.A. Malkov
```

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
```

lab13Pa.c

```
int main(){
    int pipe_fds[2];
    int read_fd, write_fd;

    pipe(pipe_fds);
    read_fd=pipe_fds[0];
    write_fd=pipe_fds[1];

    pid_t pid1, pid2;
```

```
pid1=fork();

if(pid1==0){
    close(write_fd);

    close(STDIN_FILENO);
    dup(read_fd);    //dup2(read_fd, STDIN_FILENO);

    if(execvp("./lab13a-2",NULL)==-1)
        perror("execvp call : ");

    close(read_fd);
}
else{
    pid2=fork();
```



```
if(pid2==0){  
    close(read_fd);  
  
    close(STDOUT_FILENO);  
    dup(write_fd); //dup2(write_fd, STDOUT_FILENO);  
  
    if(execvp("./lab13a-1",NULL)==-1)  
        perror("execvp call : ");  
  
    close(write_fd);  
}  
}  
  
return 0;  
}
```

Именованные каналы

```
> mkfifo ff_test
```

```
> ls -ltr
```

```
итого 32
```

-rw-r--r--	1	malkov	users	396	Dec	12	15:28	lab14Fa-c.c
-rw-r--r--	1	malkov	users	632	Dec	12	15:28	lab14Fa-s.c
-rwxr-xr-x	1	malkov	users	11864	Dec	12	15:28	lab14Fa-s
-rwxr-xr-x	1	malkov	users	11624	Dec	12	15:28	lab14Fa-c
prw-r--r--	1	malkov	users	0	Dec	13	18:22	ff_test

```
> cat < ff_test  
Hello!  
How are you?
```

```
> cat > ff_test  
Hello!  
How are you?  
See you soon.
```

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <string.h>
#define BUF_SIZE 80
```

```
int main(){
    int server_fd;
    char buf[BUF_SIZE];
    int bytes_ret;
    memset(buf,0,BUF_SIZE);
```

```
//  mkfifo("ff_test", S_IRUSR | S_IWUSR | S_IWGRP);
    server_fd = open("ff_test", O_RDONLY);
```

lab13esn.c

```
while(1){  
    fgets(buf, BUF_SIZE, fdopen(server_fd, "r"));  
    fprintf(stdout, "%s", buf);  
}  
close(server_fd);  
return 0;  
}
```

lab13ecn.c

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#define BUF_SIZE 80
```

```
int main(){
    int server_fd;
    char buf[BUF_SIZE];
    memset(buf,0,BUF_SIZE);
    //mkfifo("ff_test", S_IRUSR | S_IWUSR | S_IWGRP);
    server_fd = open("ff_test", O_WRONLY);
```

```
while(1){  
    fgets(buf, BUF_SIZE, stdin);  
    write(server_fd, buf,BUF_SIZE);  
}  
close(server_fd);  
  
return 0;  
}
```

```
~/Лекция13> ./lab13ecn  
Hi!  
Are you alright?  
See you next Monday.
```

```
/Лекция13> ./lab13esn  
Hi!  
Are you alright?  
See you next Monday.
```


Именованные каналы MS Windows

Сервер

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

np1.c

```
void main()
{
HANDLE hPipe;
LPTSTR lpPipeName = TEXT("\\\\.\\pipe\\MyPipe");
char buff[255];
DWORD iBytesToRead = 255, i;
```

```
hPipe = CreateNamedPipe(  
    lpPipeName,                // имя канала  
    PIPE_ACCESS_DUPLEX,        // чтение и запись из канала  
    PIPE_TYPE_MESSAGE |        // передача сообщений по каналу  
    PIPE_READMODE_MESSAGE |    // режим чтения сообщений  
    PIPE_WAIT,                 // синхронная передача сообщений  
    PIPE_UNLIMITED_INSTANCES,  // число экземпляров  
    4096,                      // размер выходного буфера  
    4096,                      // размер входного буфера  
    NMPWAIT_USE_DEFAULT_WAIT,  // тайм-аут клиента  
    NULL);                    // защита по умолчанию
```

```
if (hPipe == INVALID_HANDLE_VALUE) {  
    printf("CreatePipe failed: error code %d\n", (int)GetLastError());  
    return;  
}
```

```
if((ConnectNamedPipe(hPipe, NULL))==0){  
    printf("client could not connect\n");  
    return;  
}
```

```
ReadFile(hPipe, buff, iBytesToRead, &iBytesToRead, NULL);  
for(i=0; i<iBytesToRead; i++)  
    printf("%c",buff[i]);  
  
return;  
}
```

Клиент

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

np2.c

```
void main(){
    HANDLE hPipe;
    LPTSTR lpPipeName = TEXT("\\\\.\\pipe\\MyPipe");
    //TEXT("\\\\WANDERER\\pipe\\MyPipe");
    DWORD iBytesToWrite;
    char buff[] = "A message from a client";
```

```
hPipe = CreateFile(  
    lpPipeName, // имя канала  
    GENERIC_READ | GENERIC_WRITE, // чтение и запись в канал  
    0, // нет разделяемых операций  
    NULL, // защита по умолчанию  
    OPEN_EXISTING, // открытие существующего канала  
    0, // атрибуты по умолчанию  
    NULL); // нет шаблона атрибутов
```

```
WriteFile(hPipe, buff, strlen(buff), &iBytesToWrite, NULL);  
CloseHandle(hPipe);  
}
```

```
c:\>np1
```

```
A message from a client
```

```
c:\>
```

```
c:\>np2
```

```
c:\>_
```