МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

Лабораторная работа №1
по курсу «Операционные системы»

Выполнил: П. Д. Косарим
Группа: М8О-207БВ-24
Преподаватель: Е. С. Миронов

Москва, 2025

Условие

Цель работы: Приобретение практических навыков в:
- Управление процессами в ОС
- Обеспечение обмена данных между процессами посредством каналов

Задание: Составить и отладить программу на языке Си, осуществляющую работу с процессами и взаимодействие между ними в одной из двух операционных систем. В результате работы программа (основной процесс) должен создать для решение задачи один или несколько дочерних процессов. Взаимодействие между процессами осуществляется через системные сигналы/события и/или каналы (pipe). Необходимо обрабатывать системные ошибки, которые могут возникнуть в результате работы. родительского процесса пишет имя файла, которое будет передано при создании дочернего процесса. Родительский и дочерний процесс должны быть представлены разными программами. Родительский процесс передает команды пользователя через pipe1, который связан с стандартным входным потоком дочернего процесса. Дочерний процесс принеобходимости передает данные в родительский процесс через pipe2. Результаты своей работы дочерний процесс пишет в созданный им файл. Допускается просто открыть файл и писать туда, не перенаправляя стандартный поток вывода. Пользователь вводит команды вида: «число число число<endline>». Далее эти числа передаются от родительского процесса в дочерний. Дочерний процесс считает их сумму и выводит её в файл. Числа имеют тип int. Количество чисел может быть произвольным.

Вариант: 1

Метод решения

1. Родительский процесс создает два канала (pipe1, pipe2) и дочерний процесс
2. Дочерний процесс получает имя файла через аргумент командной строки
3. Взаимодействие через pipe1 (родитель → дочерний) и pipe2 (дочерний → родитель)
4. Обработка данных: дочерний процесс вычисляет сумму чисел и пишет в файл

Описание программы

parent.c - родительский процесс:
1. Создает два канала (pipe) для межпроцессного взаимодействия
2. Получает имя файла от пользователя
3. Запускает дочерний процесс
4. Передает данные в дочерний процесс через pipe1
5. Получает результаты из дочернего процесса через pipe2

child.c - дочерний процесс:
1. Получает имя файла через аргумент командной строки
2. Читает данные из стандартного ввода (перенаправленного из pipe1)
3. Вычисляет сумму чисел
4. Записывает результат в файл
5. Выводит сообщение о завершении

Используемые системные вызовы:
1. pipe() - создание неименованного канала
2. fork() - создание нового процесса
3. dup2() - дублирование файлового дескриптора

4. exec() семейство - загрузка новой программы
5. wait() - ожидание изменения состояния процесса
6. read() - чтение из файлового дескриптора
7. write() - запись в файловый дескриптор
8. exit() - завершение процесса

Результаты

При запуске программы, вводе имени файла и сторки, программа записывает сумму цифр из строки в указанный файл и выводит в консоль информацию, о том что сумма записана в файл.

Пример работы:
напишите имя файла
1.txt
введите числа
77 11
Содержимое 1.txt:
88
Вывод в консоль:
сумма записана в файл

Выводы

В ходе выполненния лабораторной работы была изучена и освоена рабоата с процессами в операционной системе с ядром Linux и управление ими через системный вызовы.
Программа корректно создает дочерний процесс, который успешно складывает числа и записывает их сумму в указанный файл, название которого передается по каналу. Обработаны ошибки, которые могут возникнуть в ходе работы программы.

Исходная программа

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/wait.h>

#define BUFFERSIZE 512

int main() {
    int pipe1[2];
    int pipe2[2];
    pid_t pid;
    char filename[BUFFERSIZE];
    FILE* file;

    printf("  \n");
    if(fgets(filename, BUFFERSIZE, stdin) == NULL) {
        perror("fgets error");
        exit(1);
    }

    filename[strcspn(filename, "\n")] = 0;

    if(pipe(pipe1) == -1 || pipe(pipe2) == -1) {
        perror("pipe error");
        exit(1);
    }
    pid = fork();
    if(pid == -1) {
        perror("fork error");
        exit(1);
    }

    if(pid == 0) {
        close(pipe1[1]);
        close(pipe2[0]);
        dup2(pipe1[0], STDIN_FILENO);
        close(pipe1[0]);
        dup2(pipe2[1], STDOUT_FILENO);
        close(pipe2[1]);
        execl("./child", "child", filename, (char*)NULL); //  main child
        perror("execl error");
        exit(1);
    } else {
        close(pipe1[0]);
        close(pipe2[1]);
        //   1 fflush(stdout);
        char buffer[BUFFERSIZE];
        printf(" \n");
        if (fgets(buffer, BUFFERSIZE, stdin) != NULL) {
            write(pipe1[1], buffer, strlen(buffer));
        }
        close(pipe1[1]);
        //   2
        int bytes;
        while((bytes = read(pipe2[0], buffer, BUFFERSIZE)) > 0) {
```

```
            fwrite(buffer, 1, bytes, stdout);

        }
        if(bytes < 0) {
            perror("read pipe2 error");
            exit(1);
        }
        close(pipe2[0]);
        wait(NULL);
        return 0;
    }fflush(stdout);
}
```

Листинг 1: parent.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/wait.h>

#define BUFFERSIZE 512

int main(int argc, char *argv[]){ // argv[0] = "child" argv[1] = filename
    if (argc != 2){
        fprintf(stderr, "using: %s filename\n", argv[0]);
        exit(1);
    }

    char buffer[BUFFERSIZE];
    FILE* file;

    file = fopen (argv[1], "w");
    if (file == NULL){
        perror("file trouble");
        exit(1);
    }
    if (fgets(buffer, BUFFERSIZE, stdin) == NULL){
        fprintf(stderr, "not data");
        fclose(file);
        exit(1);
    }

    int sum = 0;
    char* token;
    int namber;

    token = strtok(buffer, " \t\n");

    while (token != NULL){
        if (sscanf(token, "%d", &namber) == 1){
            sum += namber;
        }
        token = strtok(NULL, " \t\n");
    }

    fprintf(file, "%d", sum);
```

```
43      printf("   \n");
44      fclose(file);
45
46      return 0;
47  }
```

Листинг 2: child.c

```
      6877  execve("./parent",["./parent"],0x7ffdb75ff438 /* 49 vars */)
= 0
      6877  brk(NULL)                         = 0x61f72551e000
      6877
mmap(NULL,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) =
0x7ca09d9c6000
      6877  access("/etc/ld.so.preload",R_OK) = -1 ENOENT (Нет такого
файла или каталога)
      6877  openat(AT_FDCWD,"/etc/ld.so.cache",O_RDONLY|O_CLOEXEC) = 3
      6877  fstat(3,{st_mode=S_IFREG|0644,st_size=56203,...}) = 0
      6877  mmap(NULL,56203,PROT_READ,MAP_PRIVATE,3,0) = 0x7ca09d9b8000
      6877  close(3)                          = 0
      6877
openat(AT_FDCWD,"/lib/x86_64-linux-gnu/libc.so.6",O_RDONLY|O_CLOEXEC) = 3
      6877
read(3,"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0\0"...,832)
= 832
      6877
pread64(3,"\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"...,784,64)
= 784
      6877  fstat(3,{st_mode=S_IFREG|0755,st_size=2125328,...}) = 0
      6877
pread64(3,"\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"...,784,64)
= 784
      6877  mmap(NULL,2170256,PROT_READ,MAP_PRIVATE|MAP_DENYWRITE,3,0) =
0x7ca09d600000
      6877
mmap(0x7ca09d628000,1605632,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3
= 0x7ca09d628000
      6877
mmap(0x7ca09d7b0000,323584,PROT_READ,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0x1b0000)
= 0x7ca09d7b0000
      6877
mmap(0x7ca09d7ff000,24576,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,
= 0x7ca09d7ff000
      6877
mmap(0x7ca09d805000,52624,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,-1
= 0x7ca09d805000
      6877  close(3)                          = 0
      6877
```

```
mmap(NULL,12288,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) =
0x7ca09d9b5000
     6877  arch_prctl(ARCH_SET_FS,0x7ca09d9b5740) = 0
     6877  set_tid_address(0x7ca09d9b5a10)   = 6877
     6877  set_robust_list(0x7ca09d9b5a20,24) = 0
     6877  rseq(0x7ca09d9b6060,0x20,0,0x53053053) = 0
     6877  mprotect(0x7ca09d7ff000,16384,PROT_READ) = 0
     6877  mprotect(0x61f724181000,4096,PROT_READ) = 0
     6877  mprotect(0x7ca09da04000,8192,PROT_READ) = 0
     6877
prlimit64(0,RLIMIT_STACK,NULL,{rlim_cur=8192*1024,rlim_max=RLIM64_INFINITY})
= 0
     6877  munmap(0x7ca09d9b8000,56203)     = 0
     6877  fstat(1,{st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0),...}) =
0
     6877  getrandom("\x43\xad\x17\xb3\x4c\xa1\x17\x35",8,GRND_NONBLOCK)
= 8
     6877  brk(NULL)                         = 0x61f72551e000
     6877  brk(0x61f72553f000)               = 0x61f72553f000
     6877
write(1,"\320\275\320\260\320\277\320\270\321\210\320\270\321\202\320\265
\320\270\320\274\321\217 \321\204\320\260\320\271\320\273"...,35) = 35
     6877  fstat(0,{st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0),...}) =
0
     6877  read(0,"1.txt\n",1024)          = 6
     6877  pipe2([3,4],0)                  = 0
     6877  pipe2([5,6],0)                  = 0
     6877
clone(child_stack=NULL,flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,child_ti
= 6879
     6877  close(3 <unfinished ...>
     6879  set_robust_list(0x7ca09d9b5a20,24) = 0
     6877  <... close resumed>)              = 0
     6879  close(4 <unfinished ...>
     6877  close(6 <unfinished ...>
     6879  <... close resumed>)              = 0
     6877  <... close resumed>)              = 0
     6879  close(5)                          = 0
     6877
write(1,"\320\262\320\262\320\265\320\264\320\270\321\202\320\265
\321\207\320\270\321\201\320\273\320\260\n",26) = 26
     6879  dup2(3,0 <unfinished ...>
     6877  read(0, <unfinished ...>
     6879  <... dup2 resumed>)               = 0
     6879  close(3)                          = 0
     6879  dup2(6,1)                        = 1
     6879  close(6)                          = 0
     6879  execve("./child",["child","1.txt"],0x7ffce64885d8 /* 49 vars
```

```
*/) = 0
      6879  brk(NULL)                         = 0x5b011b9ae000
      6879
mmap(NULL,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) =
0x70b6daf16000
      6879  access("/etc/ld.so.preload",R_OK) = -1 ENOENT (Нет такого
файла или каталога)
      6879  openat(AT_FDCWD,"/etc/ld.so.cache",O_RDONLY|O_CLOEXEC) = 3
      6879  fstat(3,{st_mode=S_IFREG|0644,st_size=56203,...}) = 0
      6879  mmap(NULL,56203,PROT_READ,MAP_PRIVATE,3,0) = 0x70b6daf08000
      6879  close(3)                         = 0
      6879
openat(AT_FDCWD,"/lib/x86_64-linux-gnu/libc.so.6",O_RDONLY|O_CLOEXEC) = 3
      6879
read(3,"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0\0"...,832)
= 832
      6879
pread64(3,"\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"...,784,64)
= 784
      6879  fstat(3,{st_mode=S_IFREG|0755,st_size=2125328,...}) = 0
      6879
pread64(3,"\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"...,784,64)
= 784
      6879  mmap(NULL,2170256,PROT_READ,MAP_PRIVATE|MAP_DENYWRITE,3,0) =
0x70b6dac00000
      6879
mmap(0x70b6dac28000,1605632,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3
= 0x70b6dac28000
      6879
mmap(0x70b6dadb0000,323584,PROT_READ,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0x1b0000)
= 0x70b6dadb0000
      6879
mmap(0x70b6dadff000,24576,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,
= 0x70b6dadff000
      6879
mmap(0x70b6dae05000,52624,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,-1
= 0x70b6dae05000
      6879  close(3)                         = 0
      6879
mmap(NULL,12288,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) =
0x70b6daf05000
      6879  arch_prctl(ARCH_SET_FS,0x70b6daf05740) = 0
      6879  set_tid_address(0x70b6daf05a10)   = 6879
      6879  set_robust_list(0x70b6daf05a20,24) = 0
      6879  rseq(0x70b6daf06060,0x20,0,0x53053053) = 0
      6879  mprotect(0x70b6dadff000,16384,PROT_READ) = 0
      6879  mprotect(0x5b0105b80000,4096,PROT_READ) = 0
      6879  mprotect(0x70b6daf54000,8192,PROT_READ) = 0
```

```
      6879
prlimit64(0,RLIMIT_STACK,NULL,{rlim_cur=8192*1024,rlim_max=RLIM64_INFINITY})
= 0
      6879  munmap(0x70b6daf08000,56203)     = 0
      6879  getrandom("\x0d\xb0\x63\x43\x18\x31\x25\x13",8,GRND_NONBLOCK)
= 8
      6879  brk(NULL)                         = 0x5b011b9ae000
      6879  brk(0x5b011b9cf000)               = 0x5b011b9cf000
      6879  openat(AT_FDCWD,"1.txt",O_WRONLY|O_CREAT|O_TRUNC,0666) = 3
      6879  fstat(0,{st_mode=S_IFIFO|0600,st_size=0,...}) = 0
      6879  read(0, <unfinished ...>
      6877  <... read resumed>"77 11\n",1024) = 6
      6877  write(4,"77 11\n",6)            = 6
      6879  <... read resumed>"77 11\n",4096) = 6
      6877  close(4)                          = 0
      6877  read(5, <unfinished ...>
      6879  fstat(3,{st_mode=S_IFREG|0664,st_size=0,...}) = 0
      6879  fstat(1,{st_mode=S_IFIFO|0600,st_size=0,...}) = 0
      6879  write(3,"88",2)                 = 2
      6879  close(3)                          = 0
      6879  write(1,"\321\201\321\203\320\274\320\274\320\260
\320\267\320\260\320\277\320\270\321\201\320\260\320\275\320\260 \320\262
\321"...,40) = 40
      6879  exit_group(0 <unfinished ...>
      6877  <... read resumed>"\321\201\321\203\320\274\320\274\320\260
\320\267\320\260\320\277\320\270\321\201\320\260\320\275\320\260 \320\262
\321"...,512) = 40
      6879  <... exit_group resumed>)         = ?
      6877  write(1,"\321\201\321\203\320\274\320\274\320\260
\320\267\320\260\320\277\320\270\321\201\320\260\320\275\320\260 \320\262
\321"...,40) = 40
      6877  read(5, <unfinished ...>
      6879  +++ exited with 0 +++
      6877  <... read resumed>"",512)        = 0
      6877  ---SIGCHLD
{si_signo=SIGCHLD,si_code=CLD_EXITED,si_pid=6879,si_uid=1000,si_status=0,si_utime=0,s
---
      6877  close(5)                          = 0
      6877  wait4(-1,NULL,0,NULL)          = 6879
      6877  exit_group(0)                     = ?
      6877  +++ exited with 0 +++
```

Листинг 3: *Strace логи*