Задача 1.

Управление летательного объекта по крену

1.Постановка задачи. В качестве объекта управления выбрана упрощенная модель снаряда. Модель объекта представлена на рис. 1.

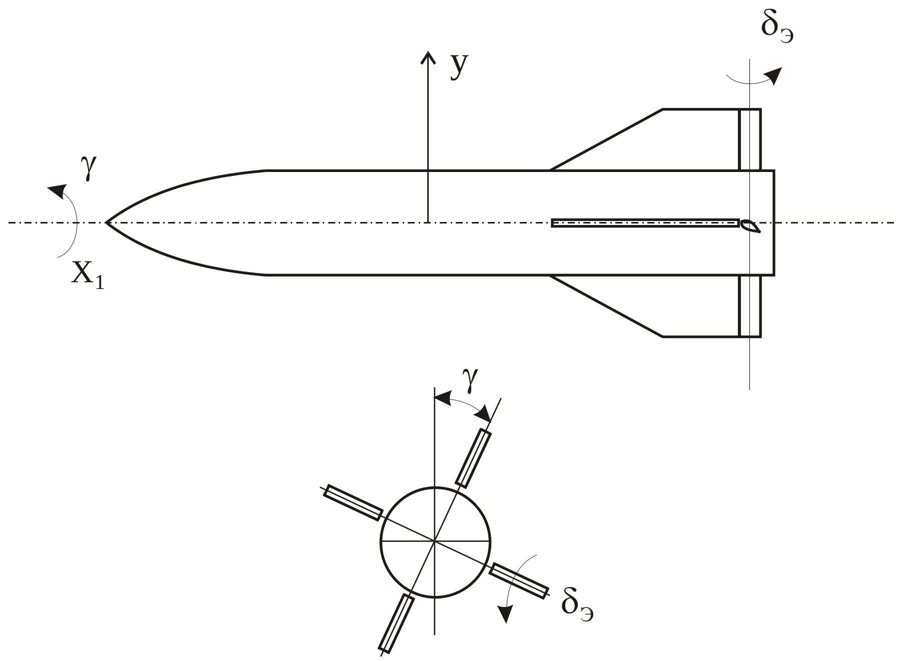

Рис. 1. Модель объекта

Движение снаряда описывается системой уравнений третьего порядка (здесь и далее временной аргумент для краткости будем опускать):

$$\begin{cases} \dot{\delta}_e = u, \\ \dot{\omega}_{x_1} = -\dfrac{1}{T_\gamma}\omega_{x_1} + \dfrac{N_e + \alpha}{T_\gamma}\delta_e, \\ \dot{\gamma} = \omega_{x_1}, \end{cases} \tag{1}$$

где

δ_e – угол отклонения элеронов,

ω_{x_1} – угловая скорость крена,

γ – угол крена,

T_γ – постоянная времени снаряда,

N_e – эффективность элеронов,

u – сигнал управления,

α – параметр возмущения ($-5 \le \alpha \le 5$).

Если принять следующие обозначения:

$$x = \begin{pmatrix} \delta_e \\ \omega_{x_1} \\ \gamma \end{pmatrix}, \; A = \begin{pmatrix} 0 & 0 & 0 \\ \dfrac{N_e}{T_\gamma} + \alpha & -\dfrac{1}{T_\gamma} & 0 \\ 0 & 1 & 0 \end{pmatrix}, \; B = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

систему (1) можно переписать в канонической форме: $\dot{x} = Ax + Bu$.

Минимизируемый функционал качества задан в виде:

$$J = \int_{t_0}^{T} \left\{ \frac{\gamma^2}{\gamma_0^2} + \frac{\delta_e^2}{\delta_{e0}^2} + \frac{u^2}{u_0^2} \right\} dt, \tag{2}$$

где

γ_0 – желаемое максимальное значение γ,

$\delta_{э0}$ – максимально допустимое значение $\delta_э$,

u_0 – максимально допустимое значение u.

Функционал J представлен в форме $\int_{t_0}^{T} (x^T Q x + u^T R u) dt$, где

$$Q = \begin{pmatrix} \dfrac{1}{\delta_{e0}^2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \dfrac{1}{\gamma_0^2} \end{pmatrix}, \; R = \frac{1}{u_0^2}.$$

Назначим числовые данные:

$$T_\gamma = 1\,с,\; N_э = 10\,c^{-1},\; u_0 = \pi\ рад \cdot c^{-1},\; \delta_{e0} = \frac{\pi}{12}\ рад,\; \gamma_0 = \frac{\pi}{180}\ рад.$$

В качестве начальных условий выберем $x^T(t_0) = \begin{pmatrix} 0 & 0 & \dfrac{\pi}{18} \end{pmatrix}$.

С учетом численных данных матрицы A, Q и R примут вид:

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 10+\alpha & -1 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \; Q = \begin{pmatrix} \left(\dfrac{12}{\pi}\right)^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \left(\dfrac{180}{\pi}\right)^2 \end{pmatrix}, \; R = \frac{1}{\pi^2}. \tag{3}$$

Построим алгоритм оптимизации заданной системы. Для начала рассмотрим случай оптимального регулятора с полной информацией о состоянии объекта, затем построим робастный регулятор с полной информацией, рассмотрим наблюдатель Луенбергера и фильтр Калмана-Бьюси без перестройки параметров. После решения перечисленных задач перейдем к конструированию алгоритма перестройки параметром для обоих наблюдателей.

2.Оптимальное управление объектом с полной информацией о его состоянии. Пусть $\alpha = 0$.Систему .1) с учетом (3) можно переписать в виде:

$$\begin{cases} \begin{pmatrix} \dot{\delta}_э \\ \dot{\omega}_{x_1} \\ \dot{\gamma} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 10 & -1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \delta_э \\ \omega_{x_1} \\ \gamma \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} u, \\ \begin{pmatrix} \delta_э(t_0) \\ \omega_{x_1}(t_0) \\ \gamma(t_0) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \dfrac{\pi}{18} \end{pmatrix}. \end{cases}$$

Функционал качества примет вид:

$$J = \int_{t_0}^{T} \left\{ \left(\frac{180}{\pi}\right)^2 \gamma^2 + \left(\frac{12}{\pi}\right)^2 \delta_e^2 + \frac{1}{\pi^2} u^2 \right\} dt .$$

Оптимальное управление определяется уравнением:

$$u(t) = -R^{-1} B^T S x(t) ,$$

где S находится из уравнения $SA + A^T S - SBR^{-1} B^T S + Q = 0$.

Состояние оптимальной системы есть решение линейного дифференциального уравнения

$$\begin{cases} \dot{x}(t) = \left[A - BR^{-1} B^T S \right] x(t), \\ x(t_0) = x_0. \end{cases}$$

Задача 2.

Напомним определение сжимающего отображения.

Пусть заданы $(X,\, d)$ — метрическое пространство и отображение $A: X \to X$. Отображение A называется сжимающим, если

$$\exists\, \gamma \in [0,1) : d(Ax,\, Ay) \le \gamma d(x,\, y) \forall x,\, y \in X.$$

Рассмотрим отображение $A: Ax = \dfrac{1}{4+x}$, $A: \mathbb{R}_+ \to \mathbb{R}_+$. Оно удовлетворяет условиям теоремы о неподвижной точке сжимающего отображения. Во-первых, действует из полного метрического пространства $\mathbb{R}_+$ в себя. Во-вторых, является сжимающим: действительно,

$$d(Ax, Ay) = |Ax - Ay| = \left|\frac{1}{4+x} - \frac{1}{4+y}\right| = \left|\frac{y-x}{(4+x)(4+y)}\right| \le$$

$$\le \frac{1}{16}|y-x| = \frac{1}{16} d(x,y)\ \forall x, y \in \mathbb{R}_+,$$

таким образом, мы нашли коэффициент $\gamma = \dfrac{1}{16}$ из определения сжимающего отображения.

Значит, согласно теореме о неподвижной точке сжимающего отображения, существует единственное решение $\tilde{x}$ уравнения $Ax = x$, решение $\tilde{x} \in \mathbb{R}_+$ называется неподвижной точкой отображения A. Оно может быть найдено как предел последовательности $x_{n+1} = Ax_n, n \in \mathbb{N} \cup \{0\}$, где x_0 произвольный элемент $\mathbb{R}_+$.

В данной задаче рассмотрена последовательность $x_0 = 0$, $x_1 = Ax_0 = [0;4] = \dfrac{1}{4}$, $x_2 = Ax_1 = [0;4,4] = \dfrac{4}{17}$ и т. д., т. е. значение цепной дроби $[0;4,4,\ldots,4,\ldots]$ по определению, данному в условии задачи, совпадает с пределом последовательности x_n, и, согласно следствию теоремы о неподвижной точке сжимающего отображения, является неподвижной точкой отображения A. Неподвижную точку находим из уравнения $x = \dfrac{1}{4+x}$, которое имеет два решения $-2+\sqrt{5}$ и $-2-\sqrt{5}$, из которых только $-2+\sqrt{5} \in \mathbb{R}_+$.

Во второй части задачи нужно оценить точность приближения цепной дроби $[0;4,4,\ldots,4,\ldots]$ дробью $[0;4,4]$, т. е. оценить $d(\tilde{x}, x_2)$.

Напомним, что точность приближения можно оценить по формуле $d(x_n,\, \tilde{x}) \le d(x_0,\, x_1)\dfrac{\gamma^n}{1-\gamma}$. Подставив в неё $d(x_1, x_0) = \dfrac{1}{4}$ и значение γ, имеем:

$$d(\tilde{x}, x_2) \leq \frac{1}{4} \frac{\left(\frac{1}{16}\right)^2}{1 - \frac{1}{16}} = \frac{1}{960}.$$

Ответ: $[0; 4, 4, \ldots, 4, \ldots] = \sqrt{5} - 2$; число $\sqrt{5} - 2$ приближается подходящей дробью $[0; 4, 4]$ с точностью $\frac{1}{960}$.

Задача 3.

Задача на основной тур

In [ ]: Дана функция на языке программирования Python:

In [ ]:
```
def recur16(n):
    if n == 0:
        return 11
    if n == 1:
        return 15
    return -recur16(n-1) + 6*recur16(n-2) - 16*n
```

Чему будет равен результат вызова `recur26(2018)` ? Диапазон определения целых чисел считать неограниченным (т.е., целые числа не переполняются), размер стека также считать неограниченным (т.е. максимальное число рекурсивных вызовов не ограничено).

Решение

Данная функция реализует неоднородное линейное рекуррентное соотношение

$$a_n = -a_{n-1} + 6a_{n-2} - 16n$$

с начальными условиями $a_0 = 11$ и $a_1 = 15$.

Общим решением неоднородного рекуррентного соотношения является сумма общего решения однородного соотношения (уже полученное в демо-туре) и частного решения неоднородного соотношения. Будем искать частное решение в виде

$$a_n = \alpha n + \beta$$

с неопределенными коэффициентами α и β. У меня получилось $\alpha = 4$ и $\beta = 11$, т.е. общее решение имеет вид

$$a_n = c_1 2^n + c_2(-3)^n + 4n + 11\ .$$

Начальные условия дают $c_1 = c_2 = 0$, и

Ответ: результатом вызова recur16(2018) будет являться число $4 * 2018 + 11 = 8083$

In [12]:
```
# Just checking the sequential verion of the recurrence :-)

def curr(prev, pprev):
    return -prev + 6*pprev

a = [11, 15]
aa = a.copy()
for j in range(11):
    n = j+2
    a += [curr(a[-1], a[-2]) - 16*n]
    aa += [4*n + 11]

print(a)
print(aa)
```

```
[11, 15, 19, 23, 27, 31, 35, 39, 43, 47, 51, 55, 59]
[11, 15, 19, 23, 27, 31, 35, 39, 43, 47, 51, 55, 59]
```

Задача 4.

В течение некоторого времени ξ ремонтная бригада ждёт заказа, $\xi \sim E(\lambda)$. Длительность обслуживания бригадой заказа - время $\eta \sim E(\mu)$. В начальный момент времени у бригады нет работы. Доходы и расходы прямо пропорциональны времени выполнения заказов и времени простоя. За выполнение заказа бригаде платят v единиц стоимости в единицу времени, за время простоя бригада теряет m единиц стоимости в единицу времени.

Построить математическую модель в виде марковского случайного процесса (пояснить, почему допустимо рассматривать процесс как марковский). Найти среднюю прибыль бригады за время T в стационарном режиме, вычислив с обоснованием предельные распределения вероятностей состояний.

Краткое решение.

$$\gamma(\mathrm{t}) = \begin{cases} 0, & \textit{если бригада простаивает в момент времени } t, \\ 1, & \textit{если бригада работает в момент времени } t \end{cases}$$

В начальный момент времени бригада ждёт заказа:

$$P_0(0) = 1 = P\{\gamma(0) = 0\}$$

Процесс – марковский (так как выполняется свойство отсутствия последействия, длительность обслуживания и длительность ожидания не зависит от того, сколько длилось до этого обслуживание/ожидание).

Найдём интенсивности перехода:

$$a_{ij} = \lim_{h\to 0}\frac{P_{ij}(h)}{h}, i \neq j; i, \mathrm{j} \in \mathrm{E}$$

$$a_i = \lim_{h\to 0}\frac{1 - P_{ii}(h)}{h}, i \in \mathrm{E}$$

Пусть ω_h - число операций, которые завершатся за время h.

$$P_0(\omega_h \geq 2) = o(h), \mathrm{h} \to 0$$

$$P_{01}(h) = P\{\gamma(\mathrm{t+h}) = 1 / \gamma(t) = 0\} = \mathrm{P}\{\xi + \eta > \mathrm{h}, \xi < \mathrm{h}\} + \mathrm{o(h)} =$$

$$= \int_0^h P\{\xi \in (\mathrm{x; x + dx})\}\mathrm{P}\{\eta > \mathrm{h - x}\} + \mathrm{o(h)} =$$

$$= \int_0^h f_\xi(x)(1 - F_\eta(h - x))\,\mathrm{dx} = \int_0^h \lambda e^{-\lambda x} e^{-\mu x} dx = \lambda e^{-\mu h}\frac{1}{\mu - \lambda}(e^{(\mu-\lambda)h} - 1) =$$

$$= \frac{\lambda}{\mu - \lambda}(e^{-\lambda h} - e^{-\mu h}) = \frac{\lambda}{\mu - \lambda}(1 - \lambda \mathrm{h} - 1 + \mu \mathrm{h} + \mathrm{o(h)}) = \lambda \mathrm{h} + o(h), h \to 0$$

$$P_{00}(h) = P\{\xi > \mathrm{h}\} + \mathrm{o(h)} = e^{-\lambda h} + o(h) = 1 - \lambda h + o(h), h \to 0$$

$$P_{10}(h) = P\{\xi + \eta > \mathrm{h}, \eta < \mathrm{h}\} + \mathrm{o(h)} = \int_0^h f_\eta(x)(1 - F_\xi(h - x))\,\mathrm{dx} =$$

$$= \int_0^h e^{-\lambda(h-x)} \mu e^{-\mu x}\,\mathrm{dx} = \mu \mathrm{h} + \mathrm{o(h)}, h \to 0$$

$P_{10}(h) = \mu \mathrm{h} + \mathrm{o(h)}, h \to 0$

$P_{11}(h) = 1 - \mu \mathrm{h} + \mathrm{o(h)}, h \to 0$

Тогда

$$a_{10} = \lim_{h\to 0}\frac{P_{10}(h)}{h} = \lim_{h\to 0}\frac{\mu h + o(h)}{h} = \mu$$

$a_1 = \mu,$

$a_0 = \lambda$

Составим дифференциальные уравнения Колмогорова для вероятностей состояний:

$P_i(t) = P\{\gamma(t) = i\}$

$$P_i(t+h) = \sum_{k\in E} P_{ki}(\mathrm{h})\,\mathrm{P}_k(\mathrm{t})$$

$$P_i(t+h) - P_i(t) = \sum_{\substack{k\in E\\ k\neq i}} P_{ki}(\mathrm{h})\,\mathrm{P}_k(\mathrm{t}) - (1 - \mathrm{P}_{ii}(h))P_i(t)$$

(Делим на h и устремляем h к нулю).

$$P_i^{'}(t+h) = \sum_{\substack{k\in E\\ k\neq i}} a_{ki}\,\mathrm{P}_k(\mathrm{t}) - a_i P_i(t), \mathrm{i} \in \mathrm{E}$$

Получили систему дифференциальных уравнений:

$$\begin{cases} P_0^{'}(t) = a_{10}P_1(t) - a_0 P_0(t) \\ P_1^{'}(t) = a_{01}P_0(t) - a_1 P_1(t) \\ P_0(0) = 1, P_1(0) = 0 \end{cases}$$

$$\begin{cases} P_0^{'}(t) = \mu P_1(t) - \lambda P_0(t) \\ P_1^{'}(t) = \lambda P_0(t) - \mu P_1(t) \\ P_0(0) = 1, P_1(0) = 0 \end{cases}$$

$P_0^{'}(t) = -(\lambda + \mu)P_0(t)$

При решении системы:

$P_0\ (t) = c(t)e^{-(\lambda+\mu)t}$

$P_0^{'}(t) = c'(t)e^{-(\lambda+\mu)t} - (\lambda+\mu)c(t)e^{-(\lambda+\mu)t}$

Получаем:

$c'(t)e^{-(\lambda+\mu)t} - (\lambda+\mu)c(t)e^{-(\lambda+\mu)t} = -(\lambda+\mu)c(t)e^{-(\lambda+\mu)t} + \mu$

$c'(t)e^{-(\lambda+\mu)t} = \mu$

$$c(t) = \frac{\mu}{\lambda+\mu} e^{(\lambda+\mu)t} + c_1$$

$$P_0(0) = 1$$

$$c_1 = \frac{\lambda}{\lambda+\mu}$$

$$\mathrm{P}_0(t) = \frac{\lambda}{\lambda+\mu} e^{-(\lambda+\mu)t} + \frac{\mu}{\lambda+\mu}$$

$$\mathrm{P}_1(t) = 1 - P_0(t) = -\frac{\lambda}{\lambda+\mu} e^{-(\lambda+\mu)t} + \frac{\lambda}{\lambda+\mu}$$

Предельное распределение:

$$\pi_i = \lim_{t\to\infty} \mathrm{P}_i(t), \mathrm{i} = 0{,}1$$

$$\pi_0 = \frac{\mu}{\lambda+\mu},$$

$$\pi_1 = \frac{\lambda}{\lambda+\mu}$$

Средний доход за время Т в стационарном режиме (t->inf)

Средний доход=$(v\pi_1 - m\pi_2)\mathrm{T}$

Задача 5.

(операционные системы, системное программирование)

Определить значения переменных *a, b, c, d, a1,b1,c1,d1,* а также содержимое массивов *buf* и *buf1* после выполнения программы на языке программирования Си в UNIX–подобной операционной системе при условии, что файла *a.txt* не существует в текущей директории. Обосновать свое решение.

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

void  main()
{ int a, b=0, c=0, d, a1, b1, c1, d1, h[2];
char buf[50], buf1[50];
buf[0]=0; buf1[0]=0;
close(0);
close(1);
pipe[h];                        //*
if(fork()==0)
{
creat("a.txt", 640);                    // (1.1)
printf("Big ");                 // (1.2)
b = write(3,"Children", 10);            // (1.3)
a = open("a.txt", 0);                   // (1.4)
c = read( a, buf, 4);                   // (1.5)
d =write( h[1], "Hello my", 5);  // (1.6)
exit(0);                                //(1.7)
}
else
{
wait(&b);                               //(2.1)
```

```
a1=dup(c);                    //(2.2)
b1=read(a1,buf1,10);          //(2.3)
c1=read(h[0],buf1,20);        //(2.4)
d1=read(b,&buf1[c1],10);      //(2.5)
f=open("a.txt",0);            //(2.6)
f1=read(f, buf, 6);           //(2.7)
}
}
```

Решение.

Сразу следует отметить, что в приведенном выше фрагменте программы оказались две лишние строки (пункты 2.6 и 2.7). Так как в задании про переменные *f* и *f1* нет вопросов, их не надо принимать во внимание. Также в п. * вместо круглых скобок указаны квадратные. Некоторые из участников олимпиады обратили на это внимание, а некоторые – не обратили. Но все, - правильно интерпретировали эту неувязку.

При запуске этой программы операционная система создаст соответствующий процесс. Во время создания процесса автоматически откроются файлы стандартного ввода, стандартного вывода и стандартного протокола с пользовательскими дескрипторами файлов 0, 1 и 2, соответственно. Процессу выделится запрашиваемая в программе память и проинициализируются значения переменных ***b, c, buf[0]*** и ***buf1[0]***.

Далее закрываются два файла с пользовательскими дескрипторами 0 и 1 (*close(0);close(1);).* Освобождаются 0-ая и 1-ая строки таблицы пользовательских дескрипторов файлов процесса (ТПДФП). С помощью системного вызова (СВ) *pipe(h)* создается межпроцессный канал, т.е. открываются два пользовательских дескриптора файлов *h[0] =0 и h[1]=1,* т.к. 0-ая и 1-ая записи таблицы пользовательских дескрипторов файлов свободны, то этими дескрипторами будут заняты эти записи.

После этого, посредством СВ *fork()* создается процесс (назовем его сын). Оба процесса и «процесс-отец» (пункт 2) и «процесс-сын» (пункт 1) начинают выполняться параллельно, причем начальные условия у них одинаковые, т.е. и текущие значения переменных, и значения дескрипторов открытых файлов. Но т.к. в «процессе-отце» первая инструкция *wait(&b),* то он приостановит свое выполнение до завершения процесса-сына.

1.1. Процесс-сын сначала с помощью СВ *creat("a.txt", 640)* создает файл в текущем каталоге *a.txt* с правами доступа *640.* Т.к. права доступа заданы в десятичной системе счисления (не *0640*), то в переводе в восьмеричную систему счисления они соответствуют *01200.* Эти права доступа разрешают владельцу файла только производить запись. Для других категорий пользователей (группы и прочих) нет никаких разрешений. Пользовательский дескриптор этого файла будет равен *3* (первая свободная строка в ТПДФП).

1.2. Далее процесс-сын выполняет функцию *printf("Big ").* При этом в файл с пользовательским дескриптором *1* должна записаться строка символов *"Big ",* но т.к. буфер вывода у процедуры *printf()* не закрыт, ОС будет ожидать закрытия буфера и записывать эту информацию повременит. Следует отметить, что файл с пользовательским дескриптором *1 – h[1]* – это межпроцессный канал.

1.3. При выполнении СВ *write(3,"Children", 10)* в файл с пользовательским дескриптором *3* должно записаться *10* байтов. Файл с пользовательским дескриптором 3 – это *a.txt*, который в силу действий СВ *creat* (пункт 1.1) уже открыт на запись (напоминаем, что СВ *creat()* не только создает файл, но и одновременно открывает его на запись). Поэтому, несмотря на указанную строку символов *Children* длиной в *7* байтов, в файл *a.txt* запишется *10* байтов, - это *7* байтов указанной строки и *3* байта из памяти процесса, которые находятся за этой строкой. Таким образом, ***b=10.***

1.4. При выполнении СВ *open("a.txt", 0)* должен открыться файл *a.txt* в режиме «для чтения». Но т.к. в правах доступа к этому файлу (п. 1.1) нет разрешения «на чтение», этот СВ не сможет выполниться, поэтому возвращаемое значение ***a=-1***.

1.5. При выполнении СВ *read(a, buf, 4)* произойдет ошибка, т.к. пользовательский дескриптор файла, из которого требуется прочитать *4* байта, *a=-1,* т.е. файл не открыт, поэтому ***c=-1***.

1.6. При выполнении СВ *write(h[1], "Hello my", 5)* в файл с пользовательским дескриптором *h[1]* (а это межпроцессный канал) будет записано *5* байтов. Т.к. СВ *write()* работает без буферизации в межпроцессный канал запишется строка *Hello,* а возвращаемое значение СВ ***d = 5***.

1.7. При выполнении функции *exit(0)* закрываются все открытые в в «процессе-сыне» файлы, а также выталкиваются все незакрытые буфера вывода, поэтому в межпроцессный канал запишется еще строка *"Big "* (пункт 1.2). «Процесс-сын» завершает свое существование.

2.1. «Процесс-отец» дождался завершения «процесса-сына» по СВ *wait(&b).* В переменную *b* запишется код завершения «процесса-сына», т.е. аргумент функции *exit().* Таким образом, ***b = 0.***

2.2. При выполнении СВ *dup()* пользовательский дескриптор файла *c* копируется на первую свободную строку ТПДФП. Т.к. значение *c=0* , этот файл открыт и является межпроцессным каналом, то СВ *dup()* выполнится и возвращаемое значение ***a1 = 3***.

2.3. При выполнении инструкции *read(a1,buf1,10)* из межпроцессного канала считывается 10 байтов в массив *buf1*. Т.к. в канале находится всего 9 байтов («*HelloBig »)* все они будут считаны и в ***buf1*** запишется «*HelloBig »*, а возвращаемое значение СВ *read()* ***b1 = 9***.

2.4. Далее выполняется СВ *read(h[0],buf1,20).* Требуется опять считать из межпроцессного канала 20 байтов в массив *buf1*. Но, т.к. канал представляет собой *fifo* файл и предыдущим СВ (2.3) из него уже была считана вся информация при выполнении этой инструкции будет получен код ответа 0, что означает конец файла, т.е. ***c1 = 0*** и значение *buf1* не изменится.

2.5. При выполнении СВ *read(b,&buf1[c1],10)* из файла с пользовательским дескриптором файла *b = 0 = h[0]* считывается 10 байтов, а т.к. из межпроцессного канала все давным-давно считано (2.3.-2.4.), то опять получим признак конца файла, т.е. ***d1 = 0***, и значение *buf1* не изменится.

Поскольку у каждого из процессов и «процесса-отца», и «процесса-сына» свое адресное пространство, результаты их выполнения можно представить в сводной таблице:

	«Процесс-сын»	«Процесс-отец»
a	***-1***	***не определен***
b	***10***	***0***
c	***-1***	***0***
d	***5***	***не определен***
a1	***не определен***	***3***
b1	***не определен***	***9***
c1	***не определен***	***0***
d1	***не определен***	***0***
buf	***пусто***	***пусто***
buf1	***пусто***	***HelloBig***