Олимпиада для студентов и выпускников – 2017 г.

Направление «Прикладная математика»
Профиль: «Прикладная математика»

Время выполнения задания – 240 мин.

Задача 1.

Пусть для задачи управления системой

$$\frac{d}{dt}x(t) = f(x(t), u(t), t)$$

с функционалом

$$J(x,u) = K(x(T),T) + \int_{t_0}^{T} L(x,u,t)dt$$

для любых начальных $x(t_0) = x_0$ и t_0 существует единственное оптимальное управление. Показать, что значение полной производной гамильтониана по времени вдоль оптимальной траектории равно частной производной гамильтониана по времени.

Решение задачи

Запишем гамильтониан

$$H(x,u,\lambda,t) = L(x,u,t) + \lambda^T(t) f(x,u,t) .$$

Найдем полную производную гамильтониана по времени

$$\frac{dH(x,u,\lambda,t)}{dt} =$$
$$= \frac{\partial H(x,u,\lambda,t)}{\partial t} + \frac{\partial H(x,u,\lambda\ ,t)}{\partial x}\frac{dx(t)}{dt} + \frac{\partial H(x,u,\lambda,t)}{\partial \lambda}\frac{d\lambda(t)}{dt} + \frac{\partial H(x,u,\lambda,t)}{\partial u}\frac{du(t)}{dt},$$

Учитывая, что

$$\frac{d}{dt}x(t) = \left\{\frac{\partial}{\partial \lambda}H(x,u,\lambda,t)\right\}^T, \quad \frac{d}{dt}\lambda(t) = -\left\{\frac{\partial}{\partial x}H(x,u,\lambda,t)\right\}^T, \quad \frac{\partial}{\partial u}H(x,u,\lambda,t) = 0,$$

получаем:

$$\frac{dH(x,u,\lambda,t)}{dt} = \frac{\partial H(x,u,\lambda,t)}{\partial t} .$$

Задача 2.

Покажите, что любое бесконечномерное замкнутое подпространство в l^2 содержит вектор не принадлежащий l^1.

Решение.

Пусть $\mathbb{X}$ — бесконечномерное замкнутое линейное подпространство в l^2. Предположим, что любой вектор из $\mathbb{X}$ принадлежит l^1. Рассмотрим множества

$$\mathbb{X}_n = \{x \in l^2 : \| x \|_{l^1} \le n\}, \quad n = 1, 2, \ldots.$$

В силу нашего предположения имеем

$$\mathbb{X} = \bigcup_{n=1}^{\infty} \mathbb{X}_n.$$

Так как $\mathbb{X}$ является полным (замкнутое подпространство полного пространства — полно), то в силу теоремы Бэра множества $\mathbb{X}_n$ не могут все быть нигде не плотными. Поэтому существует n_0 такое, что $\mathbb{X}_{n_0}$ плотно в некотором шаре в $\mathbb{X}$, т.е. в

$$B(a, r) = \{x \in \mathbb{X} : \| x - a \|_{l^2} \le r\},$$

где $a \in \mathbb{X}$ — центр шара и $r > 0$ — радиус.

Заметим, что все множества $\mathbb{X}_n$ — замкнуты в $\mathbb{X}$ (тривиальное упражнение). Поэтому $\mathbb{X}_{n_0}$ содержит шар $B(a, r)$. Пусть теперь y — произвольный вектор из $\mathbb{X}$. Считая, что $y \ne 0$, положим

$$x = a + \frac{r}{\| y \|_{l^2}} y.$$

Ясно, что $x \in B(a, r)$. Таким образом, получаем $x \in \mathbb{X}_{n_0}$. Кроме того, так как $a \in B(a, r)$, имеем $a \in \mathbb{X}_{n_0}$. Следовательно,

$$\| a + \frac{r}{\| y \|_{l^2}} y \|_{l^1} \le n_0, \qquad \| a \|_{l^1} \le n_0.$$

Из этих оценок немедленно получаем

$$\| \frac{r}{\| y \|_{l^2}} y \|_{l^1} \le 2n_0.$$

Полагая $c = 2n_0/r$, видим, что для любого $y \in \mathbb{X}$ выполняется соотношение

$$\| y \|_{l^1} \le c \| y \|_{l^2} .$$

Пусть $h_1, h_2, \ldots$ ортонормированный базис в $\mathbb{X}$. Получаем для любых чисел $c_1, c_2, \ldots$

$$\| \sum_k c_k h_k \|_{l^1} \le c \| \sum_k c_k h_k \|_{l^2} .$$

В силу равенства Парсеваля правая часть здесь равна $(\sum_k |c_k|^2)^{1/2}$, поэтому имеем

$$\| \sum_k c_k h_k \|_{l^1} \le c (\sum_k |c_k|^2)^{1/2}.$$

Следовательно при любом N

$$\| \sum_{k=1}^N c_k h_k \|_{l^1} \le c (\sum_{k=1}^N |c_k|^2)^{1/2}.$$

Пусть $e = (e^1, e^2, \ldots)$ — произвольный вектор из l^∞ с условием $\| e \|_{l^\infty} \le 1$. Получаем

$$| \sum_{j=1}^\infty \sum_{k=1}^N c_k h_k^j e^j | \le c (\sum_{k=1}^N |c_k|^2)^{1/2}.$$

Меняя порядок суммирования, видим, что

$$| \sum_{k=1}^N c_k \sum_{j=1}^\infty h_k^j e^j | \le c (\sum_{k=1}^N |c_k|^2)^{1/2}.$$

То есть для чисел

$$b_k = \sum_{j=1}^\infty h_k^j e^j, \qquad k = 1, 2, \ldots, N,$$

получаем

$$| \sum_{k=1}^N c_k b_k | \le c (\sum_{k=1}^N |c_k|^2)^{1/2},$$

что в свою очередь немедленно дает (на левую часть смотрим как на скалярное произведение векторов $(c_1, c_2, \ldots, c_N)$ и $(b_1, b_2, \ldots, b_N)$)

$$(\sum_{k=1}^N |b_k|^2)^{1/2} \le c.$$

Итак, видим, что для любого вектора $e = (e^1, e^2, \ldots)$ из единичного шара в l^∞

$$(\sum_{k=1}^N | \sum_{j=1}^\infty h_k^j e^j |^2)^{1/2} \le c.$$

В частности, полагая, например $e^j = \sin jt$, получаем

$$\sum_{k=1}^{N} \left| \sum_{j=1}^{\infty} h_k^j \sin jt \right|^2 \le c^2.$$

Следовательно,

$$\frac{1}{\pi}\int_0^{2\pi} \sum_{k=1}^{N} \left| \sum_{j=1}^{\infty} h_k^j \sin jt \right|^2 dt \le 2c^2.$$

В силу равенства Парсеваля имеем

$$\frac{1}{\pi}\int_0^{2\pi} \left| \sum_{j=1}^{\infty} h_k^j \sin jt \right|^2 dt = \sum_{j=1}^{\infty} | h_k^j |^2 = \| h_k \|_{l^2}^2 = 1,$$

Таким образом, видим, что $N \le 2c^2$, что невозможно если N достаточно велико.

Задача 3.

Ниже представлена программа на языке Python:

```
def action(max_len):
    lst = [1, 1]
    n = 0

    while len(lst) < max_len:
        lst.append(1)
        lst.append(1)
        n += 1
        for k in range(n):
            term = lst[-n-1] + lst[-n-2]
            lst.append(term)
    return lst

n = 250
result = action(n)

print len(result)
print result[n]
print result[n + 1]
```

Что выведет программа?

Решение.

Функция action создает список lst и записывает туда построчно треугольник Паскаля (треугольную таблицу из биномиальных коэффициентов). Каждая следующая строка по длине на 1 больше предыдущей и каждое число равно сумме двух расположенных над ним чисел. Зная, значение суммы арифметической прогрессии (количество элементов), можно найти номер строки треугольника, на котором цикл остановится. Для этого необходимо решить неравенство:

$$(1+\frac{n-1}{2})*n \geq 250$$

$$n \geq \frac{-1+\sqrt{2001}}{2} \ (\approx 21.87)$$

Таким образом, в треугольнике будет целиком заполнена 21 строка и цикл остановится при заполнении 22-ой строки. Количество элементов в 22-х строках равно 253 (сумма арифметической прогрессии), но так как последний элемент предыдущей строки добавляется только на следующем шаге цикла, то длина списка равна 252.

Элемент списка с индексом 250 равен $C_{21}^{19} = \frac{21!}{19! \cdot 2!} = 210$.

Элемент списка с индексом 251 равен $C_{21}^{20} = \frac{21!}{20! \cdot 1!} = 21$.

Ответы:
1) 252
2) 210
3) 21

Задача 4.

Исследуется работа современного вычислительного комплекса, предназначенного для обработки больших массивов данных (например, данных о состоянии земной атмосферы, предназначенных для составления прогнозов погоды). В дальнейшем данный вычислительный комплекс будет для краткости именоваться системой. Один массив данных рассматривается как одно требование, поступающее в систему. Отдельные массивы поступают в систему через случайные независимые, одинаково распределенные интервалы времени, имеющие экспоненциальное распределение с известным математическим ожиданием T_0.

Вычислительный комплекс состоит из двух однотипных процессоров, функционирующих независимо друг от друга. Длительность обработки одного массива данных (одного требования) в системе случайна и имеет экспоненциальное распределение с известным математическим ожиданием T_1. В состав комплекса входит внешнее запоминающее устройство (накопитель), в котором могут одновременно сохраняться не более двух массивов данных, ожидающих обработки. Массивы данных, поступающие в периоды, когда оба процессора заняты обслуживанием и места в накопителе заняты, не сохраняются и теряются.

Необходимо провести анализ рассматриваемой системы при помощи математических методов, состоящий в следующем.

1. Ввести случайный процесс, описывающий функционирование данной системы, и определить вид этого процесса.
2. Предполагая, что система функционирует в устойчивом (стационарном) режиме, найти вероятности следующих событий:

A_0 – поступающее требование застает систему свободной;

B – поступающее требование будет потеряно.

3. Обозначим через γ случайную длительность ожидания начала обработки одного массива данных, принятого в систему. Найти условное распределение случайной величины γ и соответствующее условное математическое ожидание, определяемые при условии, что в моменты поступления в системе находилось три массива данных (два на обработке и один во внешнем накопителе).

Решение.

В соответствии с условиями задачи можно считать, что отдельные массивы данных или требования поступают в данный комплекс в моменты событий простейшего потока с параметром $\lambda = T_0^{-1}$. Длительность обслуживания распределена по экспоненциальному закону с параметром $\mu = T_1^{-1}$. Таким образом, данный вычислительный комплекс можно рассматривать как систему массового обслуживания вида $M|M|2|2$, причем параметры входящего потока λ и длительности обслуживания μ являются известными. Можно использовать теорию систем массового обслуживания $M|M|n|N$.

Основной случайный процесс, описывающий функционирование систем данного вида, это процесс $\xi(t)$ – число требований (массивов данных), находящихся в данной системе в момент времени t. Этот процесс является процессом гибели и размножения (ПГР). Таким образом, для анализа данной системы можно использовать теорию процессов гибели и размножения.

Обозначим через A_k случайное событие, состоящее в том, что в системе, функционирующей в стационарном режиме, находится k требований, $k = 0,1,2,\ldots,n+N$. В рассматриваемой модели $n = 2, N = 2$. Из теории ПГР известно, что стационарные вероятности состояний выражаются формулами

$$p_k = P(A_k) = \frac{Q_k}{\sum_{i=0}^{n+N} Q_i}, k = 0,1,\ldots,n+N, (1)$$

$$где\, Q_i = \frac{\lambda_0\lambda_1\ldots\lambda_{i-1}}{\mu_1\mu_2\ldots\mu_i}, i \geq 1; Q_0 = 1, (2)$$

Величины $\lambda_i, i = 0,1,\ldots,n,n+1,\ldots,n+N-1; \mu_i, i = 1,2,\ldots,n,n+1,\ldots,n+N$ называются интенсивностями перехода случайного процесса $\xi(t)$. В процессе, описывающем систему массового обслуживания $M|M|n|N$ эти интенсивности определяются следующим образом

$$\lambda_i = \lambda, i = 0,1,\ldots,n,n+1,\ldots,n+N-1; \mu = \begin{cases} i\mu, i = 1,2,\ldots,n, \\ n\mu, i = n+1,n+2,\ldots,n+N. \end{cases} (3)$$

Из формулы (1) с учетом (2), (3) получаем

$$Q_i = \begin{cases} \dfrac{\lambda^i}{i!\,\mu^i}, & i = 1,2,\ldots,n, \\ \dfrac{\lambda^i}{n!\,\mu^i\, n^{i-n}}, & i = n+1,\ldots,n+N. \end{cases} \quad (4)$$

Если обозначать $\rho = \dfrac{\lambda}{\mu}$, то формулу (4) можно записать в виде

$$Q_i = \begin{cases} \dfrac{\rho^i}{i!}, & i = 1,2,\ldots,n, \\ \dfrac{\rho^i}{n!}, & i = n+1,\ldots,n+N. \end{cases} \quad (5)$$

В данной задаче при $n = 2, N = 2$ формулы для $Q_i, i = 0,1,2,3,4$ имеют вид

$$Q_0 = 1; Q_1 = \rho; Q_2 = \frac{\rho^2}{2!}; Q_3 = \frac{\rho^3}{2!\,2} = \frac{\rho^3}{4}; Q_4 = \frac{\rho^4}{2!\,2^2} = \frac{\rho^3}{8} (6)$$

Из(1) с учетом (6) получаем формулы для стационарных вероятностей

$$p_0 = \frac{Q_0}{Q} = \frac{1}{Q}; p_1 = \frac{Q_1}{Q} = \frac{\rho}{Q}; p_2 = \frac{Q_2}{Q} = \frac{\rho^2}{2Q};$$

$$p_3 = \frac{Q_3}{Q} = \frac{\rho^3}{2!\,2Q} = \frac{\rho^3}{4Q};\ p_4 = \frac{Q_4}{Q} = \frac{\rho^4}{2!\,2^2 Q} = \frac{\rho^4}{8Q}, (7)$$

$$где\, Q = 1 + \rho + \frac{\rho^2}{2} + \frac{\rho^3}{4} + \frac{\rho^4}{8} (8)$$

Событие A_0, состоящее в том, что поступающее требование застает систему свободной, имеет вероятность

$$p_0 = P(A_0) = \frac{1}{Q}. (9)$$

Событие B, состоящее в том, что поступающее требование будет потеряно, реализуется, если оба процесса заняты обслуживанием и два требования ожидают начала обслуживания в накопителе. Таким образом, в системе находится 4 требования: $B = A_4$. Следовательно,

$$P(B) = P(A_4) = p_4 = \frac{\frac{\rho^4}{8}}{Q} = \frac{\rho^4}{8Q} (10)$$

Теперь предположим, что в момент поступления очередного требования в системе находится три массива данных, то есть реализуется событие A_3. Иначе говоря, предполагается, что в момент поступления в систему некоторого требования в ней уже находятся три требования, два из которых обрабатываются, а одно ожидает начала обработки в накопителе. Поступившее требование займет свободное место в накопителе и станет вторым в очереди. Необходимо определить условное распределение

$$F_3(x) = P(\gamma < x \mid A_3). (11)$$

Для нахождения распределения $F_3(x)$ проведем следующее расхождение. Обозначим через t_0 момент поступления в систему рассматриваемого требования. Обозначим также через $\eta_1^{(0)}, \eta_2^{(0)}$ случайные длительности от момента t_0 до окончания обслуживания требований, находящихся в этот момент на обслуживании в первом и втором процессорах соответственно. Тогда через время $\gamma_0 = \min\ \eta_1^{(0)}, \eta_2^{(0)}$ одно из этих требований закончит обслуживаться. Таким образом, в момент $t_1 = t_0 + \gamma_0$ один из процессоров освободится, и на место обслуженного требования поступит другое, стоящее первым в очереди. При этом рассматриваемое требование, поступившее в момент t_0, продвинется в очереди на единицу, то есть станет первым в очереди. Аналогичным образом обозначим через $\eta_1^{(1)}, \eta_2^{(1)}$ случайные длительности от момента t_1 до окончания обслуживания требований, находящихся в этот момент на обслуживании в первом и втором процессорах соответственно. Тогда через время $\gamma_1 = \min\ \eta_1^{(1)}, \eta_2^{(1)}$ одно из этих требований закончит обслуживаться. Таким образом, в момент $t_2 = t_1 + \gamma_1 = t_0 + \gamma_0 + \gamma_1$ один из процессоров освободится, и на место обслуженного требования поступит то, которое в момент t_2 стоит первым в очереди, то есть именно рассматриваемое требование, поступившее в систему в момент t_0. Следовательно, если в момент t_0 в системе находилось три требования, то есть при условии, что реализовалось событие A_3, случайная длительность ожидания начала обслуживания выражается следующей формулой

$$\gamma = t_2 - t_0 = \gamma_0 + \gamma_1 (12)$$

Из приведенных выше рассуждений следует, что условное распределение $F_3(x)$, определяемое равенством (11), совпадает с распределением случайной величины γ, выраженной формулой (12):

$$F_3(x) = P(\gamma < x \mid A_3) = P(\gamma_0 + \gamma_1 < x)(13)$$

Теперь найдем распределения случайных величин γ_0, γ и их суммы. Для этого воспользуемся свойствами экспоненциального распределения. В силу основного свойства экспоненциального распределения – свойства отсутствия последействия – случайные величины $\eta_1^{(0)}, \eta_2^{(0)}$ то же самое экспоненциальное распределение с параметром μ, что и вся операция длительности обслуживания. Таким образом,

$$P\ \ \eta_1^{(0)} < x\ \ = P\ \ \eta_2^{(0)} < x\ \ = 1 - e^{-\mu x}, x \geq 0.$$

Далее, по свойству минимума случайных независимых экспоненциальных операций распределение случайной величины $\gamma_0 = \min\ \ \eta_1^{(0)}, \eta_2^{(0)}$ также является экспоненциальным с параметром, равным сумме параметров распределений величин $\eta_1^{(0)}, \eta_2^{(0)}$, то есть 2μ. Заметим, что аналогичное свойство имеет место для любого конечного числа экспоненциально распределенных величин с различными параметрами.

Итак, доказано, что

$$P(\gamma_0 < x) = 1 - e^{-2\mu x}; x \geq 0;(14)$$

при этом математическое ожидание случайной величины γ_0

$$E\gamma_0 = \frac{1}{2\mu}(15)$$

Аналогичным образом, основываясь на свойствах экспоненциального распределения, получаем

$$P\ \ \eta_1^{(1)} < x\ \ = P\ \ \eta_2^{(1)} < x\ \ = 1 - e^{-\mu x}, x \geq 0.$$

$$P\ \ \gamma_1 < x\ \ = P\ \ \min\ \ \eta_1^{(1)}, \eta_2^{(1)}\ \ < x\ \ = 1 - e^{-2\mu x}, x \geq 0.(16)$$

$$E\gamma_1 = \frac{1}{2\mu}(17)$$

Осталось воспользоваться еще одним свойством экспоненциального распределения. Известно, что сумма независимых, одинаково распределенных случайных величин, имеющих экспоненциальное распределение с параметром α, распределена по так называемому закону Эрланга. Именно, если заданы независимые случайные величины $\theta_1, \theta_2, \ldots, \theta_m$, причем $P(\theta_i < x) = 1 - e^{-\alpha x}, x \geq 0; i = 1, 2, \ldots, m$, то величина $\theta = \theta_1 + \theta_2 + \ldots + \theta_m$ имеет распределение Эрланга

$$P(\theta < x) = P(\theta_1 + \theta_2 + \ldots + \theta_m < x) = 1 - \sum_{i=0}^{m-1} \frac{(\alpha x)^i}{i!} e^{-\alpha x}, x \geq 0; m \geq 1.$$

При $m = 2$ получаем

$$P(\theta_1 + \theta_2 < x) = 1 - e^{-\alpha x} - \alpha x e^{-\alpha x}, x \geq 0.(18)$$

В рассматриваемой задаче применена формула (18). Из (13) с учетом имеем

$$F_3(x) = P\ \gamma < x \mid A_3\ = P(\gamma_0 + \gamma_1 < x) = 1 - e^{-2\mu x} - 2\mu x e^{-2\mu x}(19)$$

Условное математическое ожидание длительности ожидания начала обслуживания при условии реализации события A_3 можно найти, используя формулу (13) и свойство математического ожидания суммы случайных величин. Именно,

$$E[\gamma \mid A_3] = E[\gamma_0 + \gamma_1] = E\gamma_0 + E\gamma_1(20)$$

Из формулы (20) с учетом (15) и (17) получаем

$$E[\gamma \mid A_3] = E\gamma_0 + E\gamma_1 = \frac{1}{2\mu} + \frac{1}{2\mu} = \frac{1}{\mu}(21)$$

Исследование поставленной задачи завершено.

Задача 5.

После выполнения представленной ниже программы в UNIX-подобной операционной системе, какие значения будут у переменных ***a, b, c, d, i, p, s*** и массива ***buffer*** при условии, что файла ***a.txt*** в текущем каталоге не существует?

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

void main()
 {
        int a=2, b=3, c=0, d=5, k=2, i=0, p=2, s=2;
        char buffer[80]=0;
        close(1);
        creat("a.txt",0664);
        if(fork()==0)
        {
                a=open("a.txt",0);
                b=write(a,"bbb",4);
                c=read(a,buffer,2);
                d=write(1,"bbb",2);
                exit(0);
        }
        else
        {
                wait(&s);
                a=write(a,"bbb",8);
                open("a.txt",0);
                k=dup(1);
                i=write(k,"aa",1);
                p=read(b,buffer,8);
                buffer[p]=0;
        }
}
```

Решение

При «запуске» программы операционная система создает процесс и одновременно автоматически открывает для него файлы стандартного ввода, стандартного вывода и стандартного протокола с пользовательскими дескрипторами файлов **0, 1** и **2**, соответственно, т.е. информация об этих файлах запишется в 0-ой, 1-ой и 2-ой строках таблицы пользовательских дескрипторов файлов (ТПДФ) открытых процессом. Далее, при выполнении процесса сначала закрывается файл с пользовательским дескриптором **1** (**close(1);**). При закрытии файла строка с индексом 1 в ТПДФ освобождается.

После этого создается в текущем каталоге файл **a.txt** с правами доступа: для владельца можно читать и писать, для группы можно читать и писать, для прочих пользователей можно только читать (**creat("a.txt",0664);**). В таблице пользовательских дескрипторов файлов (ТПДФ) контекста процесса информация об этом файле разместится в первой свободной строке, т.е. строке с индексом 1. При этом необходимо учитывать, что системный вызов (СВ) **creat()** не только создает файл, но и одновременно открывает его на «запись».

На следующем этапе выполняется СВ **fork()**. Процесс разбивается на две копии («процесс-отец» и « процесс-сын»), которые начинают выполняться параллельно. Значения всех переменных и открытых файлов на начальном этапе выполнения идентичны для обоих процессов.

«Процесс-сын» сначала открывает файл **a.txt** для чтения с пользовательским дескриптором **a**. Информация об открытом файле записывается в первую свободную строку ТПДФ. Номер этой строки - **3**. Поэтому возвращаемое значение СВ open() (пользовательский дескриптор открытого файла) **a=3**. Далее выполняется СВ **write()**. По нему требуется записать 4 байта в файл с пользовательским дескриптором **a**. Но так как этот файл открыт только для чтения, возвращаемое значение **b** будет равно -1, т.е. **b=-1.** Следующая инструкция **read()** связана с чтением из файла с пользовательским дескриптором **a** в массив **buffer** 2-ух байтов. Т.к. файл пока не содержит никаких записей, при чтении из него СВ open() вернет 0, т.е. **c=0** (признак конца файла). После выполнения СВ **write(1,"bbb",2)** в файл с пользовательским дескриптором **1** (а это файл **a.txt** открытый для записи) будет записано 2 символа "**bb**", поэтому возвращаемое значение **d=2**. И, наконец, по функции **exit()** процесс-сын прекращает свое существование.

«Процесс-отец» по СВ **wait()** ждет завершения «процесса-сына». После завершения «процесса-сына» в единственный аргумент СВ **wait()** запишется код завершения «процесса-сына». Т.к. «процесс-сын» завершается по функции **exit()**, код завершения его будет равен аргументу функции **exit()**, т.е. **s=0**. Далее выполняется СВ **write()**, по которому в файл с пользовательским дескриптором **a** требуется записать 8 байтов. Для «процесса-сына» первоначальное значение **a=2**. Это число соответствует пользовательскому дескриптору открытого файла стандартного протокола. В результате выполнения этой операции в файл стандартного протокола запишется 8 байтов и возвращаемое значение СВ **write() a** станет равным 8, т.е. **a=8**. На следующем шаге выполняется СВ **open()**. Файл **a.txt** открывается для чтения. Информация об этом файле запишется в первую свободную строку ТПДФ процесса. Номер этой строки 3. Выполнение СВ **dup()** приведет к тому, что строка ТПДФ процесса с индексом 1 скопируется на первое свободное место в ТПДФ, т.е. в таблице появится строка с индексом 4. Возвращаемым значением СВ **dup()** будет индекс этой строки, т.е. **k=4**. Следующая инструкция **(i=write(k,"aa",1);)** связана с записью в файл с пользовательским дескриптором 4 символа "**a**". Это файл **a.txt**. Он был открыт на запись еще до распараллеливания процессов и в него «процесс-сын» записал уже два байта. Поэтому указатель «чтения-записи» этого файла стоит не в начале файла и к двум предыдущим символам при записи добавится еще символ "**a**", **i=1**. Последующее выполнение СВ **read()** приведет к тому, что из файла с пользовательским дескриптором b=3 (а это файл **a.txt** открытый для чтения) будет вместо 8 запрошенных байтов считано только 3 (т.к. в этом файле находится запись из 3-х символов) и записано в массив **buffer**. Поэтому возвращаемое СВ значение **p** будет равно 3. Инструкцией **buffer(p)** проставляется признак конца строки. На этом выполнение «процесса-отца» завершается.

Итоговый ответ на решение задачи приведен в нижеследующей таблице.

	«процесс-сын»	**«процесс-отец»**
a	**3**	**8**
b	**-1**	**3**
c	**0**	**0**
d	**2**	**5**
k	**2**	**4**
i	**0**	**1**
p	**2**	**3**
s	**2**	**0**
buffer	**пусто**	**bba**