

Разделяемая память

[1. Средства межпроцессной коммуникации \(IPC\)](#)

[2. Разделяемая память](#)

[3. Потоки](#)

[4. Цели и задачи](#)

[5. Порядок выполнения лабораторной работы](#)

[6. Варианты заданий \(1-15\)](#)

[7. Варианты заданий \(16-30\)](#)

[8. Варианты заданий \(31-45\)](#)

[9. Варианты заданий \(46-60\)](#)

1. Средства межпроцессной коммуникации (IPC)

Средствами межпроцессной коммуникации (IPC – Inter-Process Communication) являются сигналы, каналы, сообщения, семафоры, разделяемая память и сокеты. Процессы выполняются в собственном адресном пространстве, они изолированы друг от друга, поэтому необходимы механизмы для взаимодействия процессов, предоставляемые самой операционной системой, и, как правило, расположенные в адресном пространстве системы.

Коммуникация между процессами необходима для решения следующих задач:

- передача данных от одного процесса к другому;
- совместное использование общих данных несколькими процессами;
- синхронизация работы процессов.

Сообщения, семафоры и разделяемую память обобщенно называют System V IPC. Эти механизмы объединяются в единый пакет, потому что их

соответствующие системные вызовы обладают близкими интерфейсами, а в их реализации используются многие общие подпрограммы. Вот основные общие свойства всех трех механизмов.

1. Для каждого механизма поддерживается общесистемная таблица, элементы которой описывают всех существующих в данный момент представителей механизма (конкретные сегменты разделяемой памяти, семафоры или очереди сообщений).

2. Элемент таблицы содержит некоторый числовой ключ, который является выбранным пользователем именем представителя соответствующего механизма. Чтобы два или более процесса могли использовать некоторый механизм, они должны заранее договориться об именовании используемого представителя этого механизма.

3. Процесс, желающий начать пользоваться одним из механизмов, обращается к системе с необходимым вызовом, входными параметрами которого является ключ объекта и дополнительные флаги, а ответным параметром является числовой дескриптор, используемый в дальнейших системных вызовах подобно тому, как используется дескриптор файла при работе с файловой системой.

4. Защита доступа к ранее созданным элементам таблицы каждого механизма основывается на тех же принципах, что и защита доступа к файлам.

2. Разделяемая память

Когда процесс А посылает данные другому процессу В через канал, происходят следующие действия: данные копируются из буфера процесса А в буфер ядра, затем эти же данные копируются из буфера ядра в буфер процесса В. Механизм разделяемой памяти позволяет исключить передачу данных через ядро, предоставляя нескольким процессам доступ к одной и той же области памяти – разделяемой памяти.

Необходимо заметить, что разделяемая память имеет и недостаток по сравнению с каналами. Так как работа с разделяемой памятью может осуществляться многими процессами, присутствует вероятность одновременного изменения содержимого памяти несколькими процессами, что может привести к ошибке. Для псевдопараллельных вычислений, вероятность такого события

намного меньше, чем при действительных параллельных вычислениях. Но и в этом случае, нельзя не обращать внимания на эту отличительную особенность разделяемой памяти от остальных механизмов коммуникации процессов. Для предотвращения одновременного изменения несколькими процессами разделяемой памяти используются семафоры, с которыми вы познакомитесь, выполняя следующую работу.

Для получения информации по механизмам `ipc`, используйте команду `$ipcs`. Для получения информации только по сегментам разделяемой памяти используйте команду `$ipcs -m`.

Функции для работы с разделяемой памятью

Для использования функций работы с разделяемой памятью необходимо подключить заголовочные файлы `sys/shm.h` и `sys/ipc.h`.

int shmget (key_t key, int size, int shmflag)

Функция создает новый сегмент разделяемой памяти с ключом `key` размером `size` байт, если `shmflag` равен `IPC_CREAT`, или находит существующий сегмент с ключом `key`, если `shmflag` равен `IPC_EXCL`. Если же `shmflag` равен `IPC_CREAT | IPC_EXCL`, функция создаст новый сегмент разделяемой памяти с ключом `key`, только когда не существует другого сегмента разделяемой памяти с тем же ключом. В случае успеха функция возвращает дескриптор разделяемой памяти или отрицательное значение, в случае неудачи. Параметр `key` может быть равен `IPC_PRIVATE`, в этом случае, система сама определяет незанятый ключ для разделяемой памяти.

Параметр shmflag также может включать флаги доступа.

Флаг доступа	Значение флага доступа
0600	Разрешить чтение и запись для владельца
0400	Разрешить чтение для владельца
0200	Разрешить запись для владельца
0060	Разрешить чтение и запись для группы
0040	Разрешить чтение для группы
0020	Разрешить запись для группы
0006	Разрешить чтение и запись для всех остальных
0004	Разрешить чтение для всех остальных
0002	Разрешить запись для всех остальных

*void *shmat (int shmid, void *shmaddr, int shmflag)*

Функция подключает сегмент разделяемой памяти с указанным дескриптором shmid к виртуальной памяти процесса, вызвавшего данную функцию. Все дочерние процессы, созданные процессом-родителем после вызова функции, также будут иметь доступ к разделяемой памяти. Если shmaddr равен NULL, адрес выделяемого сегмента определяется системой, в противном случае shmaddr задает адрес для выделяемого функцией сегмента. Если shmflag равен нулю, разделяемая память будет доступна как для чтения, так и для записи. Если shmflag равен SHM_RDONLY, разделяемая память будет доступна только для чтения.

*int shmdt (void *shmaddr)*

Функция отключает сегмент разделяемой памяти с адресом shmaddr от виртуальной памяти процесса, вызвавшего данную функцию.

Пример работы с разделяемой памятью

Процесс – родитель создает сегмент разделяемой памяти и порождает дочерний процесс, процесс-родитель находит сумму первой половины элементов массива, процесс-потомок находит сумму второй половины элементов массива и записывает вычисленную сумму в разделяемую память. Родительский процесс

дожидается окончания работы потомка, и выводит окончательный результат – сумму всех элементов массива.

При такой организации алгоритма, операции с разделяемой памятью остаются безопасными и при отсутствии семафоров.

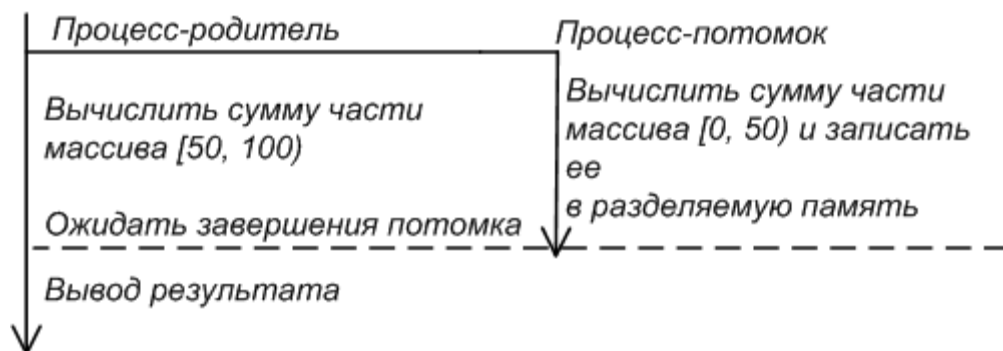


Рисунок 1. Схема процессов

```

#include <stdio.h>

#include <unistd.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <sys/wait.h>

int A[100] ; //массив, сумма элементов которого вычисляется процессами

struct mymem //структура, под которую будет выделена разделяемая память
{
    int sum ; //для записи суммы
} *mem_sum ;

int main()

{
    //запрос на создание разделяемой памяти объемом 2 байта

    //с правами чтения и записи для всех

    int shmid = shmget(IPC_PRIVATE, 2, IPC_CREAT|0666) ;

    //если запрос оказался неудачным, завершить выполнение

    if (shmid < 0 ) { fprintf(stdout, "\nОшибка") ; return 0 ; }
  
```

//теперь mem_sum указывает на выделенную разделяемую память

```
mem_sum = (mymem *)shmat(shmid,NULL,0) ;
```

//тут должна быть инициализация элементов массива A

```
int pid, sum=0 ;
```

```
pid = fork() ;
```

```
if ( pid == 0 ) //дочерний процесс
```

```
{
```

```
for (int i=0 ; i<50 ; i++) sum+=A[i] ; //вычислить сумму
```

```
mem_sum->sum=sum ; //записать ее в общую память
```

```
}
```

```
if ( pid != 0 ) //родительский процесс
```

```
{
```

```
for (int i=50 ; i<100 ; i++) sum+=A[i] ; //вычислить сумму
```

```
wait(NULL) ; //дождаться завершения процесса-потомка
```

//вывести на экран сумму всех элементов массива

```
fprintf(stdout, "\nРезультат = %d", sum+mem_sum->sum) ;
```

```
}
```

```
return 1;
```

```
}
```

3. Потоки

Потоки предоставляют возможность проведения параллельных или псевдопараллельных, в случае одного процессора, вычислений. Потоки могут порождаться во время работы программы, процесса или другого потока. Основное отличие потоков от процессов заключается в том, что различные потоки имеют различные пути выполнения, но при этом пользуются общей памятью. Таким образом, несколько порожденных в программе потоков, могут пользоваться глобальными переменными, и любое изменение данных одним потоком, будет доступно и для всех остальных. Путь выполнения потока задается при его создании, указанием его стартовой функции, созданный поток начинает выполнять команды этой функции, и завершается когда происходит возврат из функции. Любой поток завершается по окончании работы, создавшего его процесса. При создании потока, кроме стартовой функции, ему присуждается буфер для стека, определяемый программистом. Если поток в процессе своей работы превысит размерность стека, выделенного ему программистом, он будет уничтожен системой. Потоки, обладают общей памятью, операции с которой также должны защищаться семафорами.

Для создания потока используется следующая функция (заголовочный файл - sched.h).

*int clone(имя стартовой функции, void *stack, int flags, void *arg)*

Функция создает процесс или поток, выполняющий стартовую функцию, стек нового процесса/потока будет храниться в stack, параметр arg определяет входной параметр стартовой функции. Стартовая функция должна иметь такой прототип.

`int <имя функции>(void *<имя параметра>)`

Параметр flags может принимать следующие значения.

CLONE_VM – если флаг установлен, создается потомок, обладающий общей памятью с процессом-родителем (поток), если флаг не установлен, создается потомок, которому не доступна память процесса-родителя (процесс).

CLONE_FS - если флаг установлен, потомок обладает той же информацией о файловой системе, что и родитель.

CLONE_FILES - если флаг установлен, потомок обладает теми же файловыми дескрипторами, что и родитель.

CLONE_SIGHAND – если флаг установлен, потомок обладает той же таблицей обработчиков сигналов, что и родитель.

CLONE_VFORK - если флаг установлен, процесс-родитель будет приостановлен до того момента, пока не завершиться созданный им потомок.

Если флаги CLONE_FS, CLONE_FILES, CLONE_SIGHAND, CLONE_VM не установлены, потомок при создании получает копию соответствующих флагу данных от процесса-родителя. Следует понимать, что копия данных дублирует информацию от родительского процесса в момент создания потомка, но копия и данные, с которых она получена, занимают различные области памяти. Следовательно, изменение данных в процессе работы родителя, неизвестно для потомка, и наоборот.

Пример работы с потоками №1

Процесс-родитель создает поток, выполняющий функцию func.

```
#include <stdio.h>

#include <sched.h>

#include <unistd.h>

char stack[1000] ; //для хранения стека потока

int func(void * param) //стартовая функция потока

{      fprintf(stdout, "\nЗапустился поток") ;

return 1 ;

}

int main()

{      clone(func,(void*)(stack+1000-1), CLONE_VM, NULL) ; //создать поток

sleep(2) ; //заблокируем процесс-родитель, чтобы поток успел выполниться

return 1 ;

}
```


Пример работы с потоками №2

Процесс-родитель создает четыре потока, вычисляющих сумму элементов определенной части массива. Созданные потоки, выполняют функцию func, получая в качестве параметра индивидуальное целое число от 0 до 3, с помощью которого, определяются границы вычисления массива каждым потоком.

```
#include <stdio.h>

#include <sched.h>

#include <unistd.h>

#define NUMSTACK 5000 //объем стека для отдельного потока

int A[100] ; //массив, сумма элементов которого вычисляется процессами

int SUM=0 ; //для записи общей суммы

char stack[4][ NUMSTACK] ; ////для хранения стека четырех потоков

int func(void *param) //стартовая функция потоков

{

    int i, sum = 0 ; //для суммирования элементов

    //индекс массива, с которого начинается суммирование

    int p =(int *)param ; p=p*25 ;

    for(i=p ; i<p+25 ; i++) sum+=A[i] ; //вычисление суммы части элементов массива

    SUM+=sum ; //добавление вычисленного результата в общую переменную

    return 1 ;

}

int main()

{

    //тут должна быть инициализация элементов массива A

    int param[4] ; //для хранения параметров потоков

    for (int i=0 ; i<3 ; i++) //создание трех потоков
```

```

{param[i]=i ; //каждому потоку передается уникальное число

char *tostack=stack[i] ; //получить указатель на часть массива-стека потоков

//создать поток со стартовой функцией func

//первый поток получает в качестве параметра 0, второй – 1, третий - 2

        clone(func,(void*)( tostack+ NUMSTACK -1),CLONE_VM, (void *)(param+i)) ;

}

param[3]=3 ; char *tostack=stack[3] ;

        //создадим четвертый поток, указав процессу дождаться его завершения

        clone(func,(void*)( tostack+ NUMSTACK -1),

CLONE_VM|CLONE_VFORK, (void *)(param+3)) ;

fprintf (stdout,"\nРезультат = %d", SUM) ;

return 1 ;

}

```

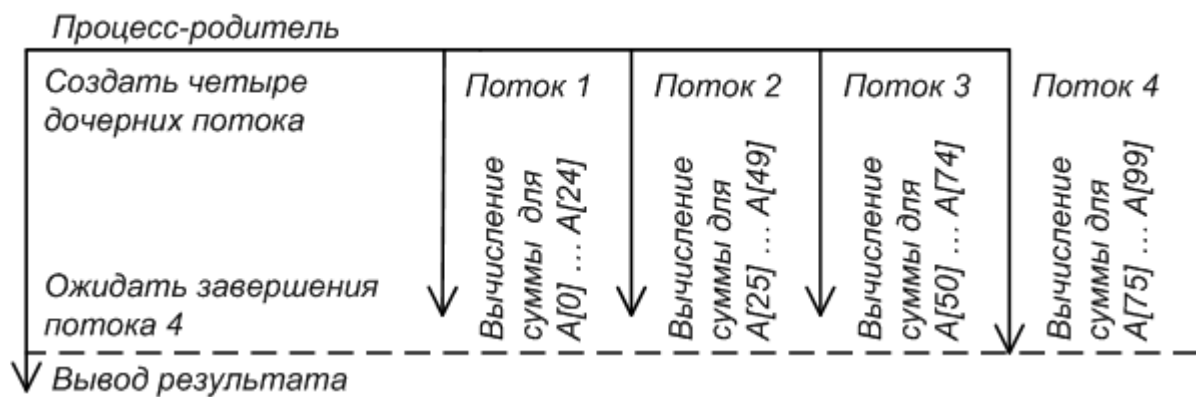


Рисунок 2. Схема потоков

4. Цели и задачи

Изучить общие принципы работы с основными средствами межпроцессной коммуникации. Научиться создавать многопроцессные алгоритмы с общей разделяемой памятью. Познакомиться с легковесными процессами – потоками.

5. Порядок выполнения лабораторной работы

1. Разработать алгоритм решения задания а, с учетом разделения вычислений между несколькими процессами. Для обмена информацией между процессами использовать разделяемую память.
2. Разработать алгоритм решения *задания б лабораторной работы 3*, используя для обмена данными не именованные каналы, а разделяемую память.
3. Разработать алгоритм решения задания б, с учетом разделения вычислений между несколькими потоками.
4. Реализовать алгоритмы решения трех задач и протестировать на нескольких примерах.
5. Посмотреть в динамике работу разделяемой памяти, используя команду `ipcs -m`.

6. Варианты заданий (1-15)

1.

а) Найти максимальный элемент из минимальных элементов в каждой строке матрицы А. Входные данные: целое положительное число n , целое положительное число k , массив чисел от А размерности $n \times k$. Использовать n или $n+1$ процессов для решения задачи.

б) Задан массив односвязных списков. Вывести на экран список, содержащий максимальное количество положительных чисел.

2.

а) Найти наибольший элемент на главной диагонали, и наименьший элемент на побочной диагонали, заменить элемент, стоящий на пересечении диагоналей на сумму двух найденных значений. Входные данные: целое положительное нечетное число n , массив чисел от A размерности $n \times n$.

б) Задан массив двусвязных списков. Вывести на экран список, не содержащий нулевых элементов.

3.

а) Определить все целые числа из интервала $[A, B]$, имеющие наибольшее количество делителей. Найти среднее арифметическое найденных чисел. Входные данные: число A , число B , целое положительное число $k > 1$ и $< (B-A)/2$. Использовать k процессов для решения задачи.

б) Охранное агентство разработало новую систему управления электронными замками. Для открытия двери клиент обязан произнести произвольную фразу из 12 слов. В этой фразе должно встречаться заранее оговоренное слово, причем только два раза и эти два слова не должны следовать друг за другом. Создать многопоточное приложение, управляющее замком.

4.

а) Найти все простые натуральные числа из интервала $[A, B]$, двоичная запись которых является палиндромом – одинаково читается как слева направо, так и справа налево. Вычислить сумму найденных чисел. Входные данные: число A , число B .

б) Задан массив односвязных списков. Составить массив средних арифметических значений каждого списка, найти сумму элементов составленного массива.

5.

а) Задана строка S, содержащая не менее двух слов, и символ с. Составит новую строку S1 из слов строки S, в которых есть символ с, и новую строку S2 из слов строки S, в которых нет символа с. Учитывать порядок вхождения слов в строку. Входные данные: строка S произвольной длины и символ с. Для решения задачи использовать столько процессов, сколько слов в строке.

б) Вычислить произведение двух квадратных матриц. Матрицы и их размерность записаны в файле.

6.

а) Задана строка S, имеющая следующий вид «число о число о число ... о число», где о может быть равно +, -, /, *. Вычислить выражение записанное в строке. Входные данные: строка S.

б) Вычислить скалярное произведение двух векторов. Вектора и их размерность задаются пользователем.

7.

а) Задана строка S, содержащая не менее двух предложений. Найти слово с максимальной длиной, встречающееся во всех предложениях или сообщить, что такого слова нет. Входные данные: строка S.

б) Задан двумерный массив. Создать многопоточное приложение для поиска суммы $\sum a_{ij}$, где a_{ij} – четные числа и $i+j$ четное число.

8.

а) Задана строка S, содержащая не менее двух слов, и символ к. Найти слово с минимальной длиной, начинающееся с символа к, или сообщить, что такого слова нет. Входные данные: строка S, символ к. Для решения задачи использовать столько процессов, сколько слов в строке.

б) Дана последовательность натуральных чисел $a_0 \dots a_{n-1}$. Создать многопоточное приложение для вычисления выражения $(a_0 - a_1)^2 + (a_2 - a_3)^2 + (a_4 - a_5)^2 + \dots$

9.

а) Определить является ли строка симметричной относительно указанного индекса r . Входные данные: строка S произвольной длины, целое число $r > 0$ и $<$ длины строки. Для решения задачи использовать два процесса.

б) Дан двумерный массив. Найти строку массива, содержащую наибольшее количество элементов, являющихся квадратами любого натурального числа.

10.

а) Задана строка S , и множество пар символов (a_i, b_i) $i = 1, 2 \dots n$, получить новую строку, заменив в строке S каждое вхождение a_i символа на b_i . Входные данные: строка S произвольной длины, целое положительное число n , множество пар символов (a_i, b_i) $i = 1, 2 \dots n$. Для решения задачи использовать четыре процесса, разделив между ними строку S .

б) Дан двумерный массив. Найти столбец массива, содержащий наибольшее количество элементов, являющихся простыми числами.

11.

а) Задана строка S , и множество пар символов (a_i, b_i) $i = 1, 2 \dots n$, получить новую строку, заменив в строке S каждое вхождение a_i символа на b_i . Входные данные: строка S произвольной длины, целое положительное число n , множество пар символов (a_i, b_i) $i = 1, 2 \dots n$. Для решения задачи использовать n процессов, работающих параллельно, причем каждый процесс находит и заменяет только свою пару символов.

б) Дан двумерный массив. Найти строку массива, содержащую наибольшее количество элементов равных минимальных элементу массива.

12.

а) Задана строка $S1$, содержащая не менее двух слов и строка $S2$, содержащая такое же количество слов, что и $S1$. Слово из строки $S2$ является синонимом соответствующего слова из строки $S1$, если оно записано с префиксом '!'. Заменить в строке $S1$ слова на их синонимы, удалив префиксы. Входные данные: строки $S1$ и $S2$, содержащие одинаковое количество слов.

б) Дан двумерный массив. Создать многопоточное приложение для поиска произведения элементов каждой строки и каждого столбца.

13.

а) Найти среднее арифметическое всех факториалов в интервале $[A, B]$. Входные данные: положительное число A , положительное число B , целое число $k \geq 2$ и ≤ 6 . Использовать k процессов для решения задачи. Предусмотреть возможность автоматического уменьшения числа процессов, если это целесообразно.

б) Дана последовательность символов $C = \{c_0 \dots c_{n-1}\}$ и символ b . Создать многопоточное приложение для определения количества вхождений символа b в строку C . Составить новую строку удалив из C все вхождения символа b .

14.

а) Сформировать одномерный массив B , элементами которого являются наибольшие из четырех рядом стоящих элементов, образующих квадрат 2×2 , в двоичном массиве A . Порядок занесения элементов в массив B не важен. Входные данные: целое положительное четное число n , массив чисел от A размерности $n \times n$.

б) Дана последовательность символов $C = \{c_0 \dots c_{n-1}\}$. Дан набор из N пар кодирующих символов (a_i, b_i) . Создать многопоточное приложение, кодирующее строку C следующим образом: строка разделяется на подстроки и каждый поток осуществляет кодирование своей подстроки.

15.

а) Задана строка S , содержащая не менее двух слов, и символ k . Составит новую строку из тех слов строки S , которые начинаются и заканчиваются символом k . Учитывать порядок расположения слов в строке. Входные данные: строка S .

б) Дан двумерный массив. Создать многопоточное приложение для поиска суммы квадратов $\sum a_{ij}^2$ в каждой строке массива. Размерность массива и его содержимое задается пользователем.

7. Варианты заданий (16-30)

16.

а) Сформировать одномерный массив В, элементами которого являются средние арифметические положительных элементов столбцов двумерного массива А. Найти сумму минимального и максимального элемента массива В. Порядок занесения элементов в массив В должен соответствовать их индексации в массиве А. Входные данные: целое положительное число n , целое положительное число k , массив чисел от А размерности $n \times k$. Использовать k или $k+1$ процессов для решения задачи.

б) Дана последовательность символов $C = \{c_0 \dots c_{n-1}\}$. Дан набор из N пар кодирующих символов (a_i, b_i) . Создать многопоточное приложение, кодирующее строку С следующим образом: поток 0 заменяет в строке С все символы a_0 на символы b_0 , поток 1 заменяет в строке С все символы a_1 на символы b_1 , и т.д. Потоки должны осуществлять кодирование последовательно.

17.

а) Сформировать массив В заменяя элементы массива А их наибольшими делителями. Найти среднее арифметическое элементов массива В, сумма индексов которых является нечетным числом. Входные данные: целое положительное число n , массив чисел от А размерности $n \times n$, целое число $k \geq 2$ и $\leq n/2$. Использовать k процессов для решения задачи.

б) Даны последовательности символов $A = \{a_0 \dots a_{n-1}\}$ и $C = \{c_0 \dots c_{k-1}\}$. В общем случае $n \neq k$. Создать многопоточное приложение, определяющее все те символы, которые входят в строку А и в строку С.

18.

а) Определить в каком числе - А или В больше вхождений цифры '0', вхождений цифры '1', вхождений цифры '2' ... вхождений цифры '9'. Определить число, сумма вхождений четных цифр которого больше. Входные данные: целое положительное число А, целое положительное число В. Для решения задачи использовать 10

процессов, каждый из которых должен определять количество вхождений своей цифры.

б) Дана квадратная матрица A , записанная в файле. Создать многопоточное приложение для поиска сумм строк и столбцов.

19.

а) Задана строка S , содержащая не менее двух целых чисел, разделителем является запятая. Составить новую строку, записав в нее все нечетные числа в обратном порядке, пример: было 2961, стало 1692. Порядок занесения чисел в новую строку не важен. Входные данные: строка S , целое число $k \geq 2$ и ≤ 8 . Использовать k процессов для решения задачи. Предусмотреть возможность автоматического уменьшения числа процессов, если это целесообразно.

б) Дан двумерный массив. Найти максимальный элемент массива, а также индексы тех строк массива, которые не содержат максимального элемента.

20.

а) Задан массив чисел, записанных в двоичной системе счисления, вычислить сумму всех чисел, не переводя их в десятичную систему счисления. Входные данные: целое положительное четное число $n > 2$, двоичный массив A размерности $n \times n$, заполненный нулями и единицами, элементы строк составляют числа, целое число $k \geq 2$ и $\leq n/2$. Использовать k процессов для решения задачи.

б) Дана квадратная матрица, записанная в файле. Определить является ли матрица магическим квадратом (суммы всех строк и столбцов одинаковы, но при этом матрица не состоит полностью из одинаковых элементов).

21.

а) Задана строка S , содержащая не менее двух слов. Необходимо найти среди слов, палиндром максимальной длины. Входные данные: строка S . Для решения задачи использовать столько процессов, сколько слов в строке. Палиндромом является фраза или слово, одинаково читаемая как слева направо, так и справа налево, пример – поп.

б) Даны результаты сдачи экзамена по группам. Создать многопоточное приложение, вычисляющее количество двоечников и отличников в каждой группе.

22.

а) Найти максимальный M и минимальный элемент m массива A и составить множество чисел, лежащих в интервале (m, M) и не содержащихся в массиве A . Входные данные: целое положительное число n , массив чисел A размерности n .

б) Дан список студентов по группам. Создать многопоточное приложение для определения количества студентов с фамилией Иванов.

23.

а) Определить какая сумма больше, сумма всех положительных чисел, стоящих выше главной диагонали и ниже второй главной диагонали, или сумма модулей всех отрицательных чисел, стоящих выше второй главной диагонали и ниже главной диагонали. Входные данные: целое положительное число n , массив чисел A размерности $n \times n$.

б) Охранное агентство разработало новую систему управления электронными замками. Для открытия двери клиент обязан произнести произвольную фразу из 10 слов. В этой фразе должно встречаться заранее оговоренное слово, причем только два раза. Создать многопоточное приложение, управляющее замком.

24.

а) Найти среднее арифметическое всех «особых» элементов матрицы A . Будем считать, элемент особым, если он больше суммы всех остальных элементов, стоящих в том же столбце. Входные данные: целое положительное число n , массив чисел A размерности $n \times n$. Использовать n или $n+1$ процессов для решения задачи.

б) Даны результаты сдачи экзамена по группам. Создать многопоточное приложение, вычисляющее общий средний балл и средний балл для каждой группы.

25.

а) Найти сумму индекса строки с максимальным элементом на главной диагонали с индексом столбца с минимальным элементом на второй главной диагонали.

Входные данные: целое положительное число n , массив чисел A размерности $n \times n$.

б) Командиру воинской части полковнику Кузнецову требуется перемножить два секретных числа. Полковник Кузнецов вызывает дежурного по части лейтенанта Смирнова и требует предоставить ему ответ. Лейтенант Смирнов будит старшего по караулу сержанта Петрова и приказывает ему предоставить ответ. Сержант Петров вызывает к себе рядового Иванова, и поручает ему ответственное задание по определению произведения. Рядовой Иванов успешно справляется с поставленной задачей, и ответ передается полковнику Кузнецову. Создать многопоточное приложение, в котором все военнослужащие от полковника до рядового моделируются потоками одного вида.

26.

а) Составить строку из максимальных и минимальных элементов строк матрицы A . Порядок элементов в строке не важен. Входные данные: целое положительное число n , целое положительное число k , массив чисел от 0 до 9 A размерности $n \times k$. Использовать n или $n+1$ процессов для решения задачи.

б) Изготовление знаменитого самурайского меча – катаны происходит в три этапа. Младший ученик мастера выковывает заготовку будущего меча. Затем старший ученик мастера закаливает меч в трех водах – кипящей, студеной и теплой. И в конце мастер собственноручно изготавливает рукоять меча и наносит узоры. Создать многопоточное приложение, в котором мастер и его ученики представлены разными потоками.

27.

а) Определить количество чисел m , являющихся квадратами некоторого целого числа, в матрице A . Заменить все простые числа в A на m . Входные данные: целое положительное число n , массив чисел A размерности $n \times n$.

б) Дана последовательность натуральных чисел $a_0 \dots a_{n-1}$. Создать многопоточное приложение для поиска суммы $\sum a_i$, где a_i – четные числа.

28.

а) Найти сумму индексов всех седловых точек матрицы A . Будем считать, элемент седловой точкой, если он является наименьшим в своей строке и наибольшим в своем столбце, либо наоборот. Входные данные: целое положительное число n , массив чисел A размерности $n \times n$.

б) Дана последовательность натуральных чисел $a_0 \dots a_{n-1}$. Создать многопоточное приложение для вычисления выражения $a_0 - a_1 + a_2 - a_3 + a_4 - a_5 + \dots$

29.

а) Найти максимальный и минимальный среди тех элементов матрицы A , сумма индексов которых равна двойке в любой целочисленной степени. Входные данные: целое положительное число n , целое положительное число k , массив чисел A размерности $n \times k$.

б) Дана последовательность натуральных чисел $a_0 \dots a_{n-1}$. Создать многопоточное приложение для поиска всех a_i , являющихся квадратами, любого натурального числа.

30.

а) Определить является ли матрица A симметричной относительно главной диагонали. Входные данные: целое положительное число n , массив чисел A размерности $n \times n$. Использовать не менее 4 процессов для решения задачи.

б) Дана последовательность натуральных чисел $a_0 \dots a_{n-1}$. Создать многопоточное приложение для поиска всех a_i , являющихся простыми числами.

8. Варианты заданий (31-45)

31.

а) В матрице A найти в каждой строке наибольший элемент и поменять его местами с элементом, стоящим на главной диагонали и в той же строке. Входные данные: целое положительное число n , массив чисел A размерности $n \times n$.

Использовать n или $n+1$ процессов для решения задачи.

б) Дана последовательность натуральных чисел $a_0 \dots a_{n-1}$. Создать многопоточное приложение для поиска минимального a_i .

32.

а) Упорядочить по возрастанию элементы в каждой строке матрицы A . Входные данные: целое положительное число n , целое положительное число k , массив чисел от A размерности $n \times k$. Использовать n или $n+1$ процессов для решения задачи.

б) Дана последовательность натуральных чисел $a_0 \dots a_{n-1}$. Создать многопоточное приложение для поиска произведения чисел $a_0 * a_1 * \dots * a_{n-1}$.

33.

а) В массиве строк хранятся фамилии и оценки учащихся. Найти учащегося с максимальным средним баллом, вывести список всех учащихся, имеющих однофамильцев. Входные данные: целое положительное число n , массив строк размерности n . Использовать n или $n+1$ процессов для решения задачи.

б) Дана последовательность символов $C = \{c_0 \dots c_{n-1}\}$ и символ b . Создать многопоточное приложение для определения количество вхождений символа b в строку C .

34.

а) Найти произведение всех элементов в матрице A , сумма или разность индексов которых является простым числом, отрицательные разности не рассматривать. Входные данные: целое положительное число n , целое положительное число k , массив чисел от A размерности $n \times k$.

б) Дана последовательность символов $C = \{c_0 \dots c_{n-1}\}$. Дан набор из N пар кодирующих символов (a_i, b_i) . Создать многопоточное приложение, кодирующее строку C следующим образом: строка разделяется на подстроки и каждый поток осуществляет кодирование своей подстроки.

35.

а) Вычислить произведение матрицы A на B , где матрица B получена из матрицы A , заменой отрицательных элементов нулем и последующим транспонированием. . Входные данные: целое положительное число n , массив чисел A размерности $n \times n$.

б) Дана последовательность натуральных чисел $a_0 \dots a_{n-1}$. Создать многопоточное приложение для поиска суммы квадратов $\sum a_i^2$.

36.

а) Двоичные числа записаны в строке, разделителем является пробел. Количество чисел равно m . Найти сумму всех двоичных чисел как двоичное число и как десятичное число. Входные данные: строка S . Для решения задачи использовать не менее m процессов.

б) Дана последовательность символов $C = \{c_0 \dots c_{n-1}\}$. Дан набор из N пар кодирующих символов (a_i, b_i) . Создать многопоточное приложение, кодирующее строку C следующим образом: поток 0 заменяет в строке C все символы a_0 на символы b_0 , поток 1 заменяет в строке C все символы a_1 на символы b_1 , и т.д. Потоки должны осуществлять кодирование последовательно.

37.

а) Проверить, можно ли составить слово S из элементов символьного массива C . Учитывать количество требуемых символов для составления слова. Входные

данные: строка S , массив символов C . Использовать для решения задачи столько процессов, сколько неповторяющихся символов в строке S .

б) Даны последовательности символов $A = \{a_0 \dots a_{n-1}\}$ и $C = \{c_0 \dots c_{k-1}\}$. В общем случае $n \neq k$. Создать многопоточное приложение, определяющее, совпадают ли посимвольно строки A и C .

38.

а) Строка содержит произвольный русский текст. Вывести сколько раз в ней встречается буква а, буква б, буква в и т.д. Найти три наиболее часто встречающиеся буквы. Входные данные: строка S . Использовать для решения задачи столько процессов, сколько букв в русском алфавите.

б) Дана квадратная матрица A . Создать многопоточное приложение для поиска сумм строк и столбцов.

39.

а) Найти максимальный по модулю элемент в матрице A и его индексы – x, y . Поменять местами строку x со строкой y и столбец x со столбцом y . Входные данные: целое положительное число n , массив чисел A размерности $n \times n$, целое положительное число $k \geq 0$ и $k \leq n-1$.

б) Дана последовательность натуральных чисел $a_0 \dots a_{n-1}$. Создать многопоточное приложение для поиска максимального a_i .

40.

а) В массиве A заменить отрицательные элементы нулями, а положительные элементы единицами. Перевести полученное двоичное число в десятичную систему счисления. Входные данные: целое положительное число n , массив чисел A размерности n , целое положительное число $k \geq 2$ и $k \leq n/2$. Использовать для решения задачи k процессов.

б) Изготовление знаменитого самурайского меча – катаны происходит в три этапа. Младший ученик мастера выковывает заготовку будущего меча. Затем старший ученик мастера закаливает меч в трех водах – кипящей, студеной и теплой. И в конце мастер собственноручно изготавливает рукоять меча и наносит узоры.

Требуется создать многопоточное приложение, в котором мастер и его ученики представлены одинаковыми потоками (обработка производится в цикле).

41.

а) Найти максимальное и минимальное целое число из интервала $[A, B]$, для которых каждое из чисел C_1, C_2, C_3 и т.д. являются делителями, учесть ситуацию когда число удовлетворяющее условиям отсутствует. Входные данные: числа A, B, K – количество чисел C и сами числа C .

б) Вычислить $\int_a^b f(x)dx$, используя метод прямоугольников. Входные данные: числа a и b , функция $f(x)$ определяется с помощью программной функции. Интервал разбить между потоками.

42.

а) Поменять местами в двумерной матрице элементы главной и побочной диагонали. Входные данные: матрица A размерности $n \times n$

б) Вывести из множества чисел произвольной размерности все тройки чисел (a, b, c) , где $c=|a-b|$ или $c=|b-a|$

43.

а) В двумерной матрице символов найти число вхождений заданного слова, при этом слово может быть записано в строке или в столбце, а также как слева направо (сверху вниз), так и наоборот.

б) Для всех чисел из множества чисел произвольной размерности определить частоту их вхождения (количество одинаковых) в множество.

44.

а) Из множества двумерных векторов найти все пары компланарных векторов. Из них найти пару векторов с максимальной обоюдной длиной. Входные данные: количество векторов и их координаты.

б) Вычислить значения функции одного переменного (функцию задать через программную функцию) на указанном интервале с указанным шагом. Проверять корректность интервала и шага.

45.

а) Вычислить сколько раз встречается число K во всей матрице и в каждой строке матрицы A . Входные данные: матрица A размерности $n \times n$, число K .

б) Вывести все двузначные числа, которые можно составить комбинацией цифр из заданного множества. Входные данные: произвольное число цифр.

9. Варианты заданий (46-60)

46.

а) Вычислить сумму тех чисел из двумерного массива A , среди соседей которых справа, слева, сверху и снизу есть хотя бы два одинаковых. Входные данные: n , массив A размерности $n \times n$.

б) Решить систему из трех линейных уравнений и трех неизвестных методом Крамера. Поток использовать для параллельного расчета необходимых определителей. Входные данные: коэффициенты системы уравнений.

47.

а) На вокзал приходят пассажиры и ожидают транспорт в определенном направлении из нескольких возможных, на вокзал приходит транспорт, увозящий пассажиров к их месту назначения. Создать многопроцессное приложение, бесконечно имитирующее систему «вокзал».

б) Найти число вхождений символов гласных букв в двумерной матрице символов.

48.

а) Найти сумму положительных чисел, расположенных ниже побочной диагонали и произведение отрицательных чисел, стоящих ниже главной диагонали. Входные данные: n , массив A размерности $n \times n$.

б) Вычислить определитель матрицы $A 4 \times 4$. Входные данные: матрица A . Разделить нахождение определителей 3×3 между несколькими потоками.

49.

а) Найти сумму особых чисел из интервала (A, B) , будем считать число особым, если в него входят ровно три одинаковых цифры, а все остальные цифры его образующие встречаются однажды. Входные данные: число A , число B .

б) Вычислить сумму, сумму квадратов и произведение элементов одномерного массива, использовать три потока, найти минимальное значение из вычисленной тройки значений.

50.

а) Найти все слова, встречающиеся в предложении дважды и только дважды. Входные данные: текстовая строка.

б) Определить три наиболее часто встречающихся символа в строке.

51.

а) Найти сумму всех трехзначных чисел, в которых хотя бы одна цифра делится нацело на цифру, стоящую в числе после нее.

б) Вычислить произведение трех матриц A, B, C . Матрицы и их размерность записаны в файле.

52.

а) Найти все строго убывающие последовательности чисел в массиве A , выделить последовательность с максимальным числом элементов. Входные данные: массив A размерности n .

б) Отсортировать массив дат (год, месяц, день) по возрастанию и по убыванию, результаты сохранить в файлы.

53.

а) Определить индексы столбцов матрицы A , числа в которых образуют строго возрастающую последовательность. Входные данные: двумерная матрица A .

б) В файле записаны студенты группы и их оценки по нескольким предметам, найти самого успешного студента, самого отстающего студента, вычислить средний балл группы, вывести список студентов, среди оценок которых есть хотя бы одна двойка.

54.

а) Определить является ли двумерная матрица особой, будем считать матрицу особой, если суммы элементов (ниже побочной диагонали и ниже главной диагонали), (выше побочной диагонали и выше главной диагонали), (ниже побочной диагонали и выше главной диагонали), (выше побочной диагонали и ниже главной диагонали) равны. Входные данные: двумерная матрица A .

б) В файле записаны названия рецептов с требуемыми ингредиентами и их количеством. Пользователь вводит произвольное число имеющихся ингредиентов, программа отображает все рецепты, которые можно приготовить из имеющихся ингредиентов.

55.

а) Найти среднее арифметическое всех тех шестизначных чисел, сумма пяти произвольных цифр которых равна K . Входные данные: число K .

б) Определить число вхождений трех указанных символов в указанной строке произвольной длины.

56.

а) Вычислить сумму значений функции $\sqrt{\sin(x)}$ на интервале $[A, B]$ с шагом h . Интервал разбить между K процессами. Входные данные: числа A, B, h, K . Если K избыточно по отношению к A, B, h , то необходимо уменьшить его автоматически.

б) Определить число одинаковых строк в матрице символов, при определении одинаковых строк учитывать сами символы и не учитывать их местоположение в строке (например «qwwee» равно «weewrq»).

57.

а) Вычислить минимальное значение функции $\sqrt{\cos(x+1)} \cdot \sqrt{\sin(x)+1}$ на интервале $[A, B]$. Входные данные: числа A, B .

б) Найти число вхождений в строке каждой цифры – 0, 1, ..., 9. Определить какая цифра встречается чаще остальных.

58.

а) Из элементов массива найти все пары чисел (a, b) , где a и b имеют хотя бы пару общих делителей (единицу как делитель не рассматривать). Выделить пару с максимальной суммой $a+b$. Входные данные: массив целых положительных чисел A размерности n .

б) Определить все строки в двумерной матрице символов, являющиеся палиндромами (например, палиндромом является строка «поп» - т.к. она одинаково читается и слева направо и справа налево).

59.

а) Составить множество чисел A_3 , таким образом, чтобы A_3 являлось пересечением множеств A_1 и A_2 . Входные данные: множества чисел произвольной размерности A_1 и A_2 , записанные в файле, внутри каждого множества каждый элемент может быть встречен лишь однажды, предусмотреть возможность корректного считывания множеств при некорректных входных данных.

б) Найти сумму квадратов нечетных элементов массива. Входные данные: целое число n , массив чисел A размерности n .

60.

а) Из целых значений, лежащих в интервале $[A, B]$ выделить такие числа, среди цифр которых цифра n встречается k раз. Найти минимальное четное значение из выделенных чисел. Входные данные: цифра n и числа k, A и B .

б) Программа запускает три потока, каждый поток случайно задает число и выводит его на экран, потоки завершаются, если хотя бы два числа равны между собой, в противном случае каждый поток снова случайно задает число и т.д.