**Практическая работа**

**«Очереди сообщений»**

**Цели и задачи**

Изучить механизм коммуникации процессов - сообщения. Научиться использовать очереди сообщений для организации взаимодействия процессов и их синхронизации.

**Общие сведения**

**Очереди сообщений**

Очередь сообщений представляет собой однонаправленный связанный список, расположенный в адресном пространстве ядра. Процессы могут записывать сообщения в очередь и изымать их из очереди. Само сообщение включает в себя тип сообщения – целое положительное число и непосредственно данные.

Рассмотрим функции для работы с очередями сообщений (подключаемые файлы – **sys/msg.h** и **sys/ipc.h**).

**int msgget (key\_t key, int msgflag)**

Функция создает новую очередь сообщений с ключом key, если msgflag равен IPC\_CREAT, или дает доступ к существующей очереди сообщений с ключом key, если msgflag равен IPC\_EXCL. Если же msgflag равен IPC\_CREAT | IPC\_EXCL, функция создаст новую очередь сообщений с ключом key, только когда не существует другой очереди с тем же ключом. Параметр shmflag также может включать флаги доступа. В случае успеха функция возвращает дескриптор очереди сообщений или отрицательное значение, в случае неудачи. Параметр key может быть равен IPC\_PRIVATE, в этом случае, система сама определяет незанятый ключ для очереди сообщений.

**size\_t msgsnd (int msgid, void \*msgp, size\_t msgsz, int msgflag)**

Функция посылает в очередь сообщений с ключом msgid сообщение msgp объемом msgsz байт. Тип сообщения не учитывается при определении объема байт. Если msgflag равен нулю процесс будет приостановлен, пока сообщениене будет помещено в очередь. Если msgflag равен IPC\_NOWAIT, процесс не ожидает помещения сообщения в очередь. Функция возвращает количество записанных байт.

**size\_t msgrcv (int msgid, void \*msgp, size\_t msgsz, long msgtype, int msgflag)**

Функция считывает сообщение в msgp из очереди с ключом msgid, максимальный объем запрашиваемого сообщения равен msgsz байт. Если объем сообщения больше, чем msgsz, сообщение не будет получено – записано в msgp. Если msgtype равен нулю из очереди будет получено первое сообщение. Если msgtype равен числу А больше нуля, из очереди будет изъято первое сообщение с типом равным А. Если msgflag равен нулю процесс будет приостановлен, пока сообщение не будет изъято из очереди. Если msgflag равен IPC\_NOWAIT, процесс не будет ожидать изъятия сообщения из очереди (если требуемое сообщение отсутствует в очереди) и продолжит выполнение. Функция возвращает количество байт в полученном сообщении.

**Пример использования очереди сообщений**

Процесс – родитель создает очередь сообщений и порождает три дочерних процесса, процесс-родитель и его потомки вычисляют сумму элементов определенной части массива, процессы-потомки записывают вычисленную сумму в очередь сообщений. Родительский процесс дожидается поступления трех сообщений, и выводит на экран окончательный результат – сумму всех элементов массива.

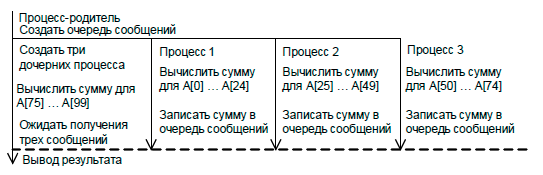


Рис.1

#include <stdio.h>

#include <unistd.h>

#include <sys/ipc.h>

#include <sys/msg.h>

int msgid ; *//для хранения дескриптора очереди сообщений*

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

struct mymsg *//структура для сообщений*

{ int mtype ; *//тип сообщения*

int mdata ; *//данные сообщения*

} m ;

int summa (int p) *//для вычисления суммы части элементов массива*

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

int begin =25\*p ; *//индекс массива, с которого начинается суммирование*

int end = begin+25 ; *//индекс массива, на котором завершается суммирование*

for(i=begin ; i<end ; i++) sum+=A[i] ; *//вычисление суммы части элементов массива*

m.mtype = 1 ; *//установить тип сообщения в 1*

m.mdata=sum ; *//записать вычисленную сумму в сообщение*

msgsnd(msgid, &m, 2, 0) ; *//послать сообщение в очередь, объем 2 байта*

return sum ; *//возвратить вычисленную сумму*

}

int main()

{//тут должна быть инициализация элементов массива А

*// создать очередь сообщений*

msgid = msgget(IPC\_PRIVATE, IPC\_CREAT|0666) ;

*//если не удалось создать очередь сообщений, завершить выполнение*

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

for (int i=0 ; i<3 ; i++) *//создать три процесса-потомка*

{

if ( fork() == 0 ) *//истинно для дочернего процесса*

{ summa(i) ; return 1 ;}

} //родительский процесс вычисляет последнюю четверть массива

int rez = summa(3) ;

for (int i=0 ; i<3 ; i++) *//дождаться получения трех сообщений*

{msgrcv(msgid, &m, 2, 0, 0) ; *//тип получаемого сообщения не важен*

rez += m.mdata ; *//добавить данные сообщения к результату*

}

*//вывести на экран сумму всех элементов массива*

fprintf(stdout,"\nСумма = %d",rez) ;

return 1;

}

**Порядок выполнения практической работы**

1.Разработать алгоритм решения задания, с учетом разделения вычислений между несколькими процессами. Для обмена информацией между процессами использовать очередь сообщений.

2.Реализовать алгоритм решения задания и протестировать его на нескольких примерах.

3.Посмотреть в динамике работу семафоров для созданного приложения, используя команду ipcs -q.

**Варианты заданий**

1.Сформировать массив, элементами которого являются целые числа больше А, меньше В и не равные С1, С2, … Сn. Найти среднее арифметическое элементов полученного массива. Входные данные: число А, число Б, произвольное количество чисел С1, С2, … Сn. Использовать не менее четырех процессов для решения задачи.

2.Из трех матриц А, В, С определить матрицу с наибольшим определителем. Найти сумму элементов для выбранной матрицы. Входные данные: массивы чисел А, В, С размерности 4x4. Использовать не менее трех процессов для решения задачи.

3.Из элементов массива найти все пары чисел (а, б), где а равно сумме всех делителей числа б, за исключением единицы и самого б. Найти пару с максимальным значением а+б. Входные данные: целое положительное число n, целое положительное число к>1, массив целых положительных чисел А размерности n. Использовать к процессов для решения задачи. Предусмотреть возможность автоматического уменьшения числа процессов, если это целесообразно.

4.Из элементов массива найти все пары чисел (а, б), где б-а=к. Найти пару с максимальным значением |а|\*|б|. Входные данные: целое положительное число n>3, целое число к, массив целых чисел А размерности n. Использовать не менее четырех процессов для решения задачи.

5.Найти сумму всех элементов массива А, которые являются числами Армстронга. Натуральное число называется числом Армстронга, когда сумма его цифр, возведенная в степень равную числу этих цифр, равна самому числу. Входные данные: целое положительное число n, целое положительное число к>1, массив натуральных чисел А размерности n. Использовать к процессов для решения задачи. Предусмотреть возможность автоматического уменьшения числа процессов, если это целесообразно.

6.Найти все четырехзначные числа abcd, для которых a+b+c+d = k или a-b+c-d = k или ab – cd = к. Вычислить сумму найденных чисел. Входные данные: целое положительное число к. Использовать девять процессов для решения задачи, где каждый процесс работает со своим числовым интервалом.

7.Найти все четырехзначные числа abcd, для которых a+b+c+d = k или a\*b\*c\*d = k или aс – bd = к. Вычислить среднее арифметическое найденных чисел. Входные данные: целое положительное число к. Использовать три процесса для решения задачи, где каждый процесс вычисляет собственное условие, из трех заданных.

8.Найти максимальный элемент в матрице А, и поменять его местами с максимальным элементом матрицы В. Затем вычислить сумму определителей полученных матриц. Входные данные: массивы чисел А, В размерности 4x4.

9.Найти все n-значные числа, цифры которых образуют убывающую (960) или возрастающую (1258) последовательность. Вычислить среднее арифметическое найденных чисел. Входные данные: целое положительное число n<11. Использовать n или n+1 процессов для решения задачи.

10.Найти все n-значные числа, делящиеся нацело на каждую из своих цифр. Вычислить среднее арифметическое найденных чисел. Входные данные: целое положительное число n<11. Использовать девять процессов для решения задачи, где каждый процесс работает со своим числовым интервалом.

11.Из элементов массива А найти все числа, равные n\*(n+1)\*(n+2), где n – любое натуральное число. Вычислить сумму максимального и минимального найденного числа. Входные данные: целое положительное число к, массив натуральных чисел А размерности к.

12.Найти процент счастливых билетов с шестизначными номерами из интервала (А, В). Входные данные: шестизначное число А, шестизначное число Б>А, целое положительное число к>1. Использовать к процессов для решения задачи. Предусмотреть возможность автоматического уменьшения числа процессов, если это целесообразно.

13.Найти все n-значные числа, содержащие к единиц и не содержащие нулей. Вычислить среднее арифметическое найденных чисел. Входные данные: целое положительное число n<11, целое неотрицательное число к≤n. Предусмотреть обработку ситуаций, когда порождение дочерних процессов является излишним.

14.Сформировать новый массив из соответствующих элементов массива А, отняв от каждого элемента сумму его цифр, от получившегося числа снова отнять сумму его цифр и так далее к раз. Найти сумму минимального элемента массива А и максимального элемента сформированного массива. Входные данные: целое положительное число к, целое положительное число n, массив целых чисел А размерности n.

15.Вычислить сумму тех целых чисел из интервала (А, В), которые равны двойке в произвольной целой степени. Входные данные: натуральное число А, натуральное число Б>А

16.Найти все числа из элементов массива А, равные произведению двух произвольных простых чисел. Вычислить сумму из к наибольших найденных чисел. Входные данные: целое положительное число к, целое положительное число n, массив целых чисел А размерности n.

17.Из множества векторов найти вектор с максимальной длиной и вектор с минимальной длиной, вычислить векторное и скалярное произведения найденных векторов. Входные данные: целое положительное число n, массив целых чисел А размерности nх3, где каждый вектор записан в строке. Использовать два процесса для решения задачи.

18.Вычислить сумму тех целых чисел из интервала (А, В), которые равны n + произвольное простое число . Входные данные: натуральное число А, натуральное число Б>А, целое число n, натуральное число к>1. Использовать к процессов для решения задачи. Предусмотреть возможность автоматического уменьшения числа процессов, если это целесообразно.

19.Из трех матриц А, В, С определить матрицу, содержащий максимальный элемент и матрицу, содержащую минимальный элемент. Вычислить произведение двух найденных матриц. Входные данные: массивы чисел А, В, С размерности 3x3. Использовать не менее трех процессов для решения задачи.

20.Найти количество вхождений в строку S, каждого символа, в ней содержащегося. Определить наиболее часто повторяющийся символ. Входные данные: строка S.