

Лабораторная работа №1

Сдать до 20.02

Тема: «Создание потоков».

Глобальные переменные не использовать!

Задача. Написать программу для консольного процесса, который состоит из двух потоков: `main` и `worker`.

Поток **main** должен выполнить следующие действия:

1. Создать массив целых чисел, размерность и элементы которого вводятся с консоли (или сгенерировать случайно).
2. Ввести время для остановки и запуска потока `worker`.
3. Создать поток `worker`, передать в поток данные: размер массива, массив и т.д.
4. Приостановить поток **worker** (`SuspendThread`), затем через некоторое время снова запустить поток.
5. Уметь создавать поток командой `_beginthreadex`
6. Дождаться завершения потока `worker`.
7. Вывести на консоль результат работы потока `worker`
8. Завершить работу.

Поток **worker** должен выполнить следующую работу (**только пункт с индивидуальным номером**):

1. Найти среднее арифметическое значение элементов массива. После каждого суммирования элементов «спать» 12 миллисекунд. Завершить свою работу.
2. Найти минимальный элемент массива. Завершить свою работу.
3. Найти сумму элементов. После каждого суммирования элементов «спать» 20 миллисекунд. Завершить свою работу.
4. Найти максимальный элемент массива. Завершить свою работу.
5. Найти количество нулевых значений. После каждого действия элементов «спать» 12 миллисекунд. Завершить свою работу.
6. Ввести новый элемент X. Найти количество элементов массива, равных X. Завершить свою работу.
7. Ввести новый элемент X. Найти количество элементов массива, больших X. Пред началом поиска «спать» 200 миллисекунд. Завершить свою работу.
8. Ввести новый элемент X. Найти элементы массива, меньшие X. Завершить свою работу.
9. Найти сумму квадратов элементов. После каждого суммирования элементов «спать» 200 миллисекунд. Завершить свою работу.
10. Ввести максимальный элемент из отрицательных элементов массива. После поиска «спать» 100 миллисекунд. Завершить свою работу.
11. Ввести минимальный элемент из положительных элементов массива. Завершить свою работу.
12. Ввести количество четных элементов из элементов массива. Завершить свою работу.
13. Ввести элементы из отрезка $[a, b]$. Завершить свою работу.
14. Ввести количество нечетных элементов из элементов массива. Завершить свою работу.
15. Ввести количество элементов кратных 3 из элементов массива. Завершить свою работу.
16. Ввести количество элементов кратных 5 из элементов массива. Завершить свою работу.
17. Ввести количество элементов кратных 9 из элементов массива. Завершить свою работу.
18. Ввести новый элемент X - вещественный. Найти количество элементов массива, целая часть которых совпадает с целой частью X. Завершить свою работу.
19. Найти сумму квадратных корней элементов. После каждого суммирования элементов «спать» 200 миллисекунд. Завершить свою работу.

Примечания.

1. Для ожидания завершения работы потока `worker` использовать функцию:

```
DWORD WaitForSingleObject(  
    HANDLE    hHandle,           // дескриптор объекта  
    DWORD     dwMilliseconds     // интервал ожидания в миллисекундах  
);
```

где второй параметр установить равным `INFINITE`. Например

```
WaitForSingleObject(hThread, INFINITE);    // ждать завершения потока
```

Здесь `hThread` – дескриптор потока `worker`.

2. Для засыпания использовать функцию:

```
VOID Sleep(  
    DWORD     dwMilliseconds     // миллисекунды  
);
```

Например, `Sleep(12);` // спать 12 миллисекунд

Дополнительное (или штрафное после 20.02) задание:

- a. Добавить третий поток **Count**;
- b. Создать поток **Count** в потоке **main**, в подвешенном состоянии.
- c. Запустить поток **Count**.

Поток **Count** выполняет:

Выводит на консоль числа фибоначчи, по возрастанию.

Лабораторная работа №2.

Тема: «Создание процессов».

Сдать до 6.03

Задание:

Написать программы двух консольных процессов **Parent** и **Child**, которые выполняют следующие действия.

Два проекта (процессы) хранить в одном **Solution** (Решении)!

В **Solution** (Решении) настроить, что бы .exe файлы лежали в одном **Debug**!

Процесс **Parent**:

- Согласно **индивидуальным вариантам** выполняет :

- Ввести размер массива, ввести элементы массива;
- Для вариантов 1,4, 6, 8, 9 – ввести необходимые дополнительные значения согласно варианту (А,В,Х,К);
- Формирует командную строку, которая содержит информацию об размерности массива, элементах и т.д. (согласно индивидуальному варианту);
- Для консоли дочернего процесса устанавливает визуальные настройки, согласно индивидуальным вариантам:
 1. Установить любой цвет текста (не белый) для **Child**.
 2. Установить ширину окна для **Child**.
 3. Установить высоту окна для **Child**.
 4. Установить ширину буфера для **Child**.
 5. Установить высоту буфера для **Child**.
 6. Установить ширину (X) смещения от верхнего левого угла экрана.
 7. Установить высоту (Y) смещения от верхнего левого угла экрана.
 8. Установить любой цвет фона (не черный) для **Child**.
 9. Установить любой цвет текста (не белый) для **Child**.
 10. Установить ширину окна для **Child** или заголовок окна.
 11. Установить высоту окна для **Child** или заголовок окна.
 12. Установить ширину буфера для **Child**.
 13. Установить высоту буфера для **Child**.
 14. Установить ширину (X) смещения от верхнего левого угла экрана.
 15. Установить высоту (Y) смещения от верхнего левого угла экрана.
 16. Установить любой цвет фона (не черный) для **Child**.
 17. Установить любой цвет текста (не белый) для **Child**.
 18. Установить ширину окна для **Child** или любой цвет фона (не черный) для **Child**.
 19. Установить высоту окна для **Child** или заголовок окна.
- Запускает дочерний процесс **Child**, которому через командную строку передается информация об размерности массива, элементах и т.д. (согласно варианту);

Процесс **Child**:

- Согласно **индивидуальным вариантам** **Child** выполняет:

1. Выполнить суммирование элементов итогового массива до заданной позиции **K**. Полученный массив вывести. Тип элементов - вещественные числа.
2. Найти в массиве повторяющиеся элементы (разместить их группы в массиве слева, остальные (одиночные) - соответственно справа). Полученный массив вывести. Тип элементов - вещественные числа.
3. Сортировка методом “пузырька”. Полученный массив вывести. Тип элементов - вещественные числа двойной точности
4. Поиск в массиве элементов из диапазона **[A,B]** (разместить их в массиве слева, остальные элементы массива - заполнить нулями). Полученный массив вывести. Тип элементов - целые числа без знака.
5. Сортировка выбором. Полученный массив вывести. Тип элементов - символы.
6. Поиск в массиве элементов **>A** (разместить их в массиве слева, остальные элементы массива - заполнить нулями). Полученный массив вывести. Тип элементов - целые числа.
7. Поиск в массиве простых чисел (разместить их в массиве слева, остальные элементы массива - справа). Полученный массив вывести. Тип элементов - целые числа без знака.

8. Поиск в массиве элементов =X (разместить их в массиве слева, остальные элементы массива - справа). Полученный массив вывести. Тип элементов - символы.
9. Выполнить произведение элементов (без 0) итогового массива до заданной позиции K. Полученный массив вывести. Тип элементов - целые числа, без знака.
10. Поиск в массиве лексем, (разделители – цифры). Полученные лексемы поместить в массиве слева, разделитель - пробел, остальные элементы - заполнить символом '0'. Полученный массив вывести. Тип элементов массива - символы.
11. Приведение массива к палиндрому (получившейся палиндром поместить в массиве слева, а лишние элементы соответственно – справа). Полученный массив вывести. Тип элементов - символы
12. Сортировка вставками. Полученный массив вывести. Тип элементов - целые числа.
13. Сортировка Шелла. Полученный массив вывести. Тип элементов - вещественные числа.
14. Поиск в массиве чисел кратных 3. (разместить их в массиве слева, остальные элементы массива - справа). Полученный массив вывести. Тип элементов - целые числа.
15. Поиск в массиве чисел кратных 5. (разместить их в массиве слева, остальные элементы массива - справа). Полученный массив вывести. Тип элементов - целые числа без знака.
16. Сортировка Хоара. Полученный массив вывести. Тип элементов - целые числа.
17. Сортировка Подсчетом. Полученный массив вывести. Тип элементов - целые числа.
18. Сортировка бинарная. Тип элементов - целые числа.
19. Поиск в массиве элементов >0 (разместить их в массиве слева, остальные элементы массива - заполнить нулями). Полученный массив вывести. Тип элементов - целые числа.

Примечания.

Для ожидания завершения работы процесса Child использовать функцию:

```
DWORD WaitForSingleObject(
    HANDLE          hHandle,          // дескриптор объекта
    DWORD dwMilliseconds // интервал ожидания в миллисекундах
); где второй параметр установить равным INFINITE, например

WaitForSingleObject(hProcess, INFINITE); // ждать завершения процесса
```

Здесь hProcess – дескриптор процесса Child.

В Solution (Решении) настроить, что бы .exe файлы лежали в одном Debug!

Дополнительное (или штрафное после 6.03) задание:

1. завершить процесс с помощью функции TerminateProcess
2. завершить процесс Parent с помощью функции ExitProcess;
3. Запустить 2-й процесс **Count** из **Parent**. У процесса **Count** менять приоритет. Процесс **Count** выводит на консоль числа фибоначчи, по возрастанию.