**Пример функции потока для задания лабораторной работы**

Условия использования рассматриваемой функции потока:

g\_dwThreadId – глобальный массив с идентификаторами запущенных потоков. Индекс 1 соответствует потоку 1, индекс 2 соответствует потоку 2.

Функция задержки:

void for\_delay(int param)

{ double x,y,z; x=2.0; y=3.0;   
 for (int i =0;i<param;i++) {z=x\*y-1; z=z+1;}

}

В качестве параметра функции потока передается структура THREAD\_PARAM

struct THREAD\_PARAM

{

int Num; */ / Номер потока*

UINT XPos; */ / Позиция X области вывода*

UINT YPos; */ / Позиция Y области вывода*

HWND hWnd; */ / Дескиптор окна вывода*

};

Предварительно создаются два объекта типа THREAD\_PARAM:

THREAD\_PARAM ThrParam1 = {1, g\_uXPos, g\_uYPos, hWnd};

THREAD\_PARAM ThrParam2 = {2, g\_uXPos, g\_uYPos+24, hWnd};.

Адреса этих объектов передаются в качестве параметра при создании потоков.

*/ / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / /*

*/ /*

*/ / Функция, выполняемая в каждом отдельном потоке (функция потока)*

*/ /*

*/ / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / /*

DWORD WINAPI ThreadFunc1(PVOID pvParam)

{

THREAD\_PARAM \* ThrParam;*/ / Локальная переменная для хранения переданного параметра*

ThrParam = (THREAD\_PARAM \*) pvParam;

#define MAXCHAR 100

TCHAR CreepingLine[MAXCHAR];*/ / Буфер для символов бегущей строки*

TCHAR buf[MAXCHAR]={0};*/ / Рабочий буфер для циклического сдвига строки*

int iBeginningIndex;*/ / Индекс начала выводимой последовательности символов*

int StringLength=0;*/ / Длина строки*

RECT rc;

HDC hDC;

int cRun=0; */ / Счетчик “пробегов” строки*

int N=5; */ / Количество “пробегов” в серии*

*/ / Формирование текста бегущей строки*

wsprintf( CreepingLine,

TEXT("40321,Петров П.П.,поток № %d, идентификатор - 0x%.8X**.** ###"),

ThrParam->Num, g\_dwThreadId[ThrParam->Num] );

*/ / Длинна строки*

StringLength = iBeginningIndex = lstrlen(CreepingLine);

lstrcpy(buf,CreepingLine);

*/ / Задание прямоугольной области вывода*

GetClientRect(ThrParam->hWnd, &rc);

rc.top = ThrParam->YPos;

rc.left = ThrParam->XPos;

rc.right= rc.right - ThrParam->XPos;

*/ / Получение контекста для вывода строки*

hDC = GetDC(ThrParam->hWnd);

*/ / Бесконечный цикл вывода строк сериями по N строк*

while(TRUE)

{

*/\* // Взаимное исключение одновременного вывода*

*// серии строк более чем одним потоком*

WaitForSingleObject(g\_hMutex, INFINITE);

*\*/*

cRun=0;

while(cRun < N-1)*// Цикл вывода серии из N строк*

{

*/ / Цикл однократного продвижения строки от последнего*

*/ / символа до первого (перемещение слева направо в области вывода)*

for(int j=0; j< StringLength; j++)

{

if(iBeginningIndex==0)

{

iBeginningIndex=StringLength;

*cRun++; / / Подсчет количества полных пробегов строки*

}

*/ / Cдвиг символов в рабочем буфере на одну позицию*

TCHAR c;

c=buf[StringLength];

for (int i=StringLength; i>0; i--)*// Цикл сдвига*

buf[i]=buf[i-1];

buf[0]=c;

*/ / Ввывод строки*

DrawText(hDC, buf,-1,&rc,DT\_LEFT|DT\_SINGLELINE);

iBeginningIndex--;

Sleep(200);*/ / приостановка потока на 200 мсек - замедление цикла*

*/ / Альтернативный" подход - замедление "вращения" цикла*

*/ / без отказа от остатка кванта путем выполнения for\_delay(mywait).*

*/ / Позволяет наблюдать эффект от изменения приоритетов потоков.*

*/ / for\_delay(mywait) - любая работа, занимающая процессор*

*/ / на некоторое время*

} */ / Конец цикла полного однократного “пробега” строки*

}*/ / Конец цикла вывода серии строк*

***/\* // конец критического участка кода – вывод серии строк***

**ReleaseMutex(g\_hMutex);**

***\*/***

}

return 0;

}