Функции библиотеки Memory.dll

1. Функции получения данных:
   1. Прием одной переменной:

int \_stdcall Get\_Char (char\* input, int Adr, int Region); - 8 бит

int \_stdcall Get\_Short (short\* input,int Adr, int Region); - 16 бит

int \_stdcall Get\_Long (long\* input, int Adr, int Region); - 32 бита

int \_stdcall Get\_Float (float\* input,int Adr, int Region); - плавающая точка

int \_stdcall Get\_Double (double\* input,int Adr, int Region); - плавающая точка двойной точности

int \_stdcall Get\_BaseVar (double\* input, int Adr); - базовая переменная(всегда double)

* 1. Прием массива переменных, размером, определяемым аргументом Size:

int \_stdcall Get\_CharBuff (char\* input, int Adr, int Region , int Size); - 8 бит

int \_stdcall Get\_ShortBuff (short\* input,int Adr, int Region , int Size); - 16 бит

int \_stdcall Get\_LongBuff (long\* input, int Adr, int Region , int Size); - 32 бита

int \_stdcall Get\_FloatBuff (float\* input,int Adr, int Region , int Size); - плавающая точка

int \_stdcall Get\_DoubleBuff(double\* input,int Adr, int Region , int Size); - плавающая точка двойной точности

int \_stdcall Get\_BaseVarBuff(double\* InBuff, int Adr, int Size); **-** базовая переменная(всегда double)

1. Функции передачи данных
   1. Передача одной переменной

input – адрес источника;

Adr - номер ячейки приемника в области памяти;

Region – определяет область, в которую записывается переменная:

0 – переменные;

1 – уставки;

int \_stdcall Set\_Char (char\* input, int Adr, int Region); - 8 бит

int \_stdcall Set\_Short (short\* input,int Adr, int Region); - 16 бит

int \_stdcall Set\_Long (long\* input, int Adr, int Region); - 32 бита

int \_stdcall Set\_Float (float\* input, int Adr, int Region); - плавающая точка

int \_stdcall Set\_Double (double\* input, int Adr, int Region); - плавающая точка двойной точности

int \_stdcall Set\_BaseVar (double\* input, int Adr); **-** базовая переменная(всегда double)

* 1. Передача массива переменных, размером, определяемым аргументом Size:

int \_stdcall Set\_CharBuff (char\* input, int Adr, int Region , int Size); - 8 бит

int \_stdcall Set\_ShortBuff (short\* input,int Adr, int Region , int Size); - 16 бит

int \_stdcall Set\_LongBuff (long\* input, int Adr, int Region , int Size); - 32 бита

int \_stdcall Set\_FloatBuff (float\* input, int Adr, int Region , int Size); - плавающая точка

int \_stdcall Set\_DoubleBuff(double\* input, int Adr, int Region, int Size); - плавающая точка двойной точности

int \_stdcall Set\_BaseVarBuff(double\* InBuff, int Adr, int Size); **-** базовая переменная(всегда double)

1. Служебные функции:
   1. Получить размеры массивов памяти различных типов данных в области переменных:

void \_stdcall Get\_SizeMBuff(int\* ByteSize, int\* WordSize, int\* LwordSize, int\* FloatSize, int\* DoubleSize);

где: ByteSize – адрес размера массива 8-ми битовых переменных (кол-во байт);

WordSize – адрес размера массива 16-ти битовых переменных (кол-во слов);

LwordSize– адрес размера массива 32 битовых переменных (кол-во длинных слов);

FloatSize -адрес размера массива переменных с плавающей точкой (кол-во переменных);

DoubleSize-адрес размера массива переменных с плавающей точкой двойной точностью (кол-во переменных);

* 1. Получить размеры массивов памяти различных типов данных в области уставок:

void \_stdcall Get\_SizeSBuff(int\* ByteSize, int\* WordSize, int\* LwordSize, int\* FloatSize, int\* DoubleSize);

где: ByteSize – адрес размера массива 8-ми битовых уставок (кол-во байт);

WordSize – адрес размера массива 16-ти битовых уставок (кол-во слов);

LwordSize– адрес размера массива 32 битовых уставок (кол-во длинных слов);

FloatSize -адрес размера массива уставок с плавающей точкой (кол-во уставок);

DoubleSize-адрес размера массива уставок с плавающей точкой двойной точностью (кол-во уставок);

* 1. Получить размер массива памяти базовых переменных:

void \_stdcall Get\_SizeBBuff(int\* Size);

где: Size – адрес размера массива (кол-во переменных);

* 1. Отправить идентификатор приложения для получения командного слова:

void \_stdcall SendHWND(HWND input);

Отправляет библиотеке идентификатор приложения input, которое будет принимать к исполнению командное слово, отправленное с помощью функции SendCtrlCode. Библиотека отправляет командное слово всем приложениям, идентификаторы которых были определены данной функцией. Может быть идентифицировано максимум 5 приложений.

* 1. Отправить командное слово приложениям

void \_stdcall SendCtrlCode(long Code);

где: Code – командное слово;

коды команд:

1 – начать отработку программы;

2 – пауза;

3 – перейти на следующий шаг программы;

4 – закрыть дебагер;

5 - перейти на указанный адрес программы(еще не реализовано, но предполагается);

0xff – сброс (обнуление памяти и переход на начальный адрес программы).