LABWORK – 1

Objective: Creating a process with using Win32 API. Creating parent and child processes.

C*ontaining some functions and structures required to create a function.*
CreateProcess function :
http://msdn.microsoft.com/en-us/library/windows/desktop/ms682425(v=vs.85).aspx
STARTUPINFO structure :
http://msdn.microsoft.com/en-us/library/windows/desktop/ms686331(v=vs.85).aspx
PROCESS_INFORMATION structure :
http://msdn.microsoft.com/en-us/library/windows/desktop/ms684873(v=vs.85).aspx
ZeroMemory function :
http://msdn.microsoft.com/en-us/library/windows/desktop/aa366920(v=vs.85).aspx

How to Create a Process in Windows by using <windows.h>
By completing the introduction to win32 api part, you will be able to understand creating process in 5 steps.

How to Create a Process in Windows by using <windows.h>
1. Initialization: the basic components of process

2. Loading to memory:

a. Defining the starting address and the length of the process

b. Free the memory as required space to avoid various overflow issues

3. Creating Process: Process structure should be defined with required parameters

4. Handling: Specify the process in all functions that perform operations on the process object

5. Termination: Terminate the process with desired exit status

EXAMPLE 1:

// build and run this example

```
#include <windows.h>
#include <stdio.h>
#include <tchar.h>

void _tmain( int argc, TCHAR *argv[] )
{
   STARTUPINFO si;
   PROCESS_INFORMATION pi;

   ZeroMemory( &si, sizeof(si) );
   si.cb = sizeof(si);
   ZeroMemory( &pi, sizeof(pi) );

   if( argc != 2 )
   {
      printf("Usage: %s [cmdline]\n", argv[0]);
      return;
   }

   // Start the child process.
   if( !CreateProcess( NULL,   // No module name (use command line)
      argv[1],       // Command line
      NULL,          // Process handle not inheritable
      NULL,          // Thread handle not inheritable
      FALSE,         // Set handle inheritance to FALSE
      0,             // No creation flags
      NULL,          // Use parent's environment block
      NULL,          // Use parent's starting directory
      &si,           // Pointer to STARTUPINFO structure
      &pi )          // Pointer to PROCESS_INFORMATION structure
   )
   {
      printf( "CreateProcess failed (%d).\n", GetLastError() );
      return;
   }

   // Wait until child process exits.
   WaitForSingleObject( pi.hProcess, INFINITE );

   // Close process and thread handles.
   CloseHandle( pi.hProcess );
   CloseHandle( pi.hThread );
}
```

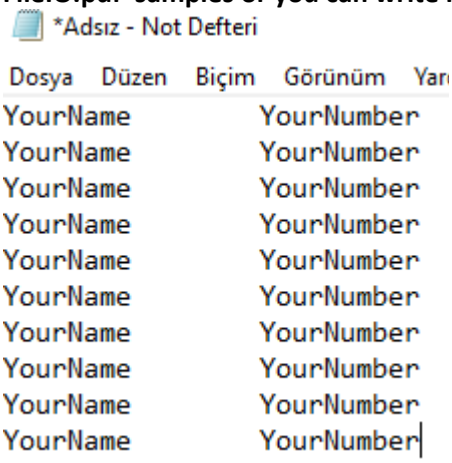
EXAMPLE 2:

// build and run this example

```
#include <windows.h>
#include <string.h>
#include <stdio.h>


int main(void)
  {
        BOOL bRet;
        STARTUPINFO si;
        PROCESS_INFORMATION pi;
        ZeroMemory(&si,sizeof(si));
        si.cb=sizeof(si);
        ZeroMemory(&pi,sizeof(pi));

         bRet=CreateProcess(NULL,"notepad.exe",NULL,NULL,FALSE,0,NULL,NULL,&si,&pi);
        if(bRet==FALSE) {
        printf("Error: %u\n",GetLastError());
        return 1;
        }

        CloseHandle(pi.hProcess);
        CloseHandle(pi.hThread); // close handles to kernel objs

  return 0;
         }
```

YOUR TASK:

1.Write and build your C program which creates a txt file and write into your name and your number 10 times. (You can use FileIO.pdf samples or you can write it on your own).



2. **And use yourprogram.exe file in another process in createProcess method as parameter.**
   Example:
   bRet=CreateProcess(NULL,"yourprogram.exe",NULL,NULL,FALSE,0,NULL,NULL,&si,&pi);

3. **Finally you should submit two C file**
   1 yourprogram.c (which creates a txt and write into your name and your number 10 times.)
   2 mainprogram.c

You can use DevC++ or Visual Studio
DevC++ : https://sourceforge.net/projects/orwelldevcpp/
Visual Studio: https://visualstudio.microsoft.com/tr/thank-you-downloading-visual-studio/?sku=Community&rel=16#