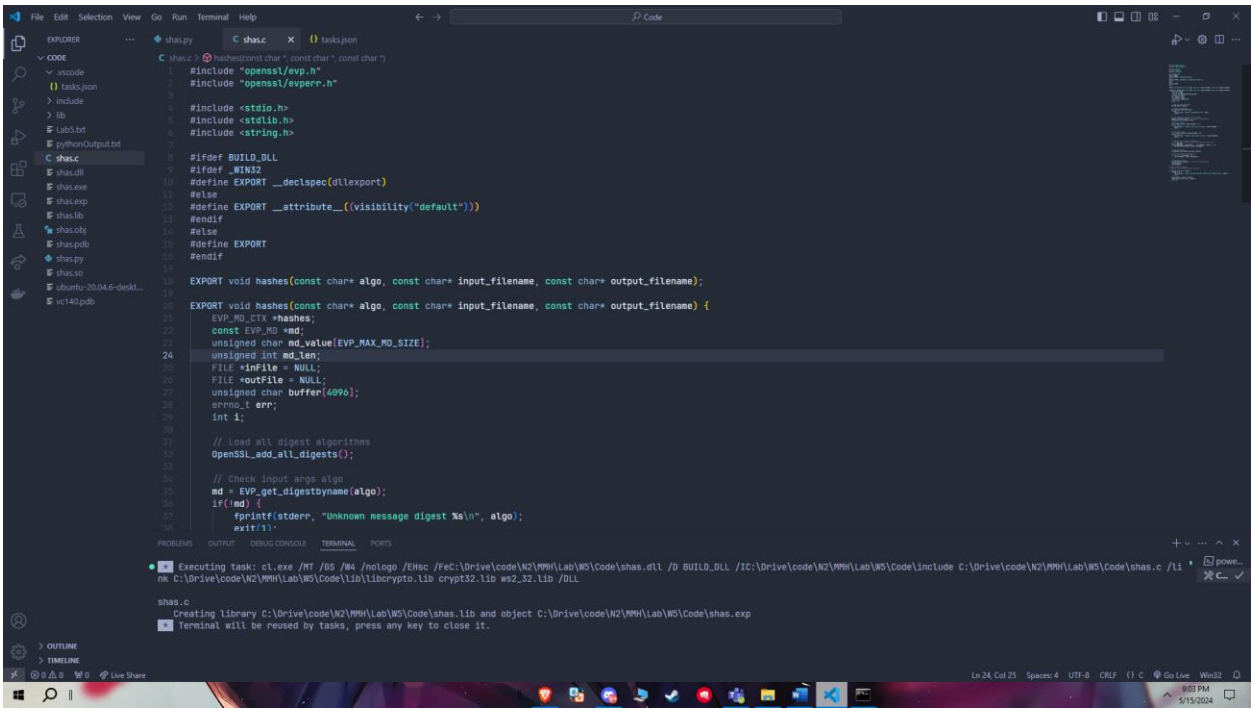
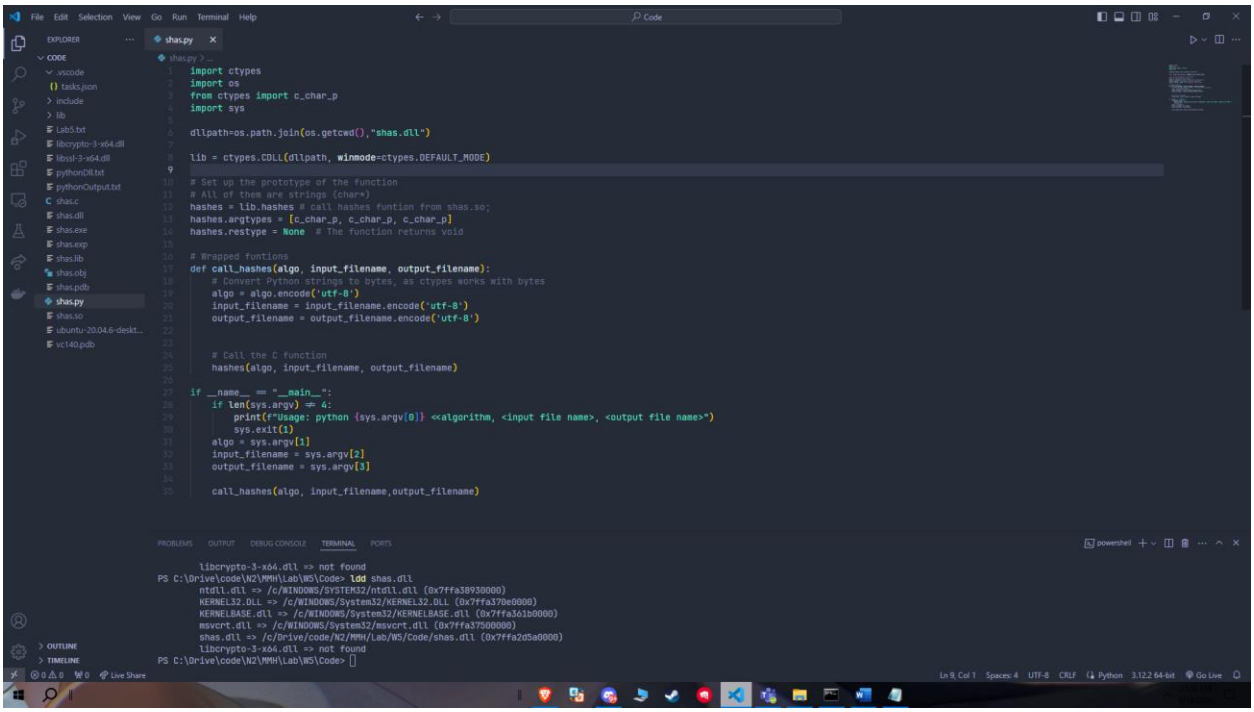
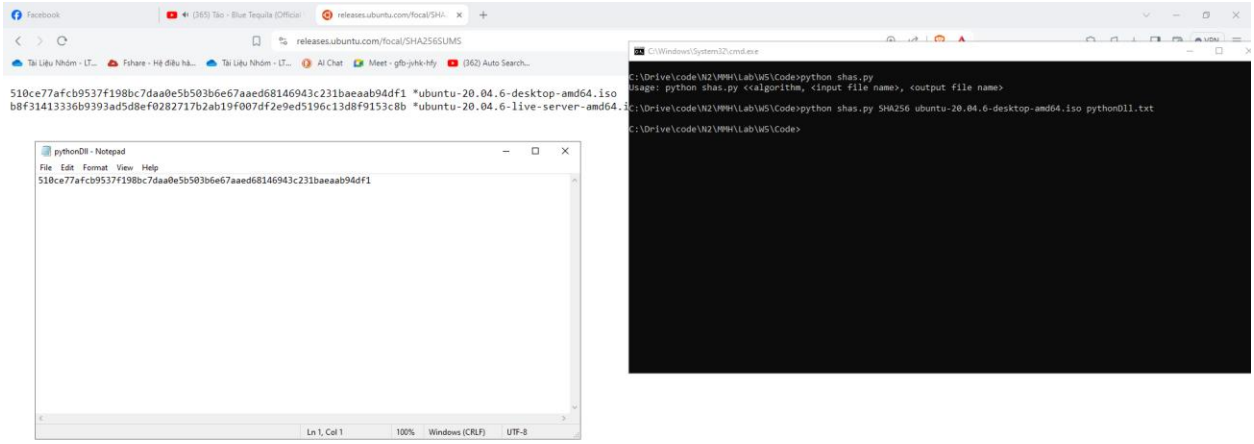
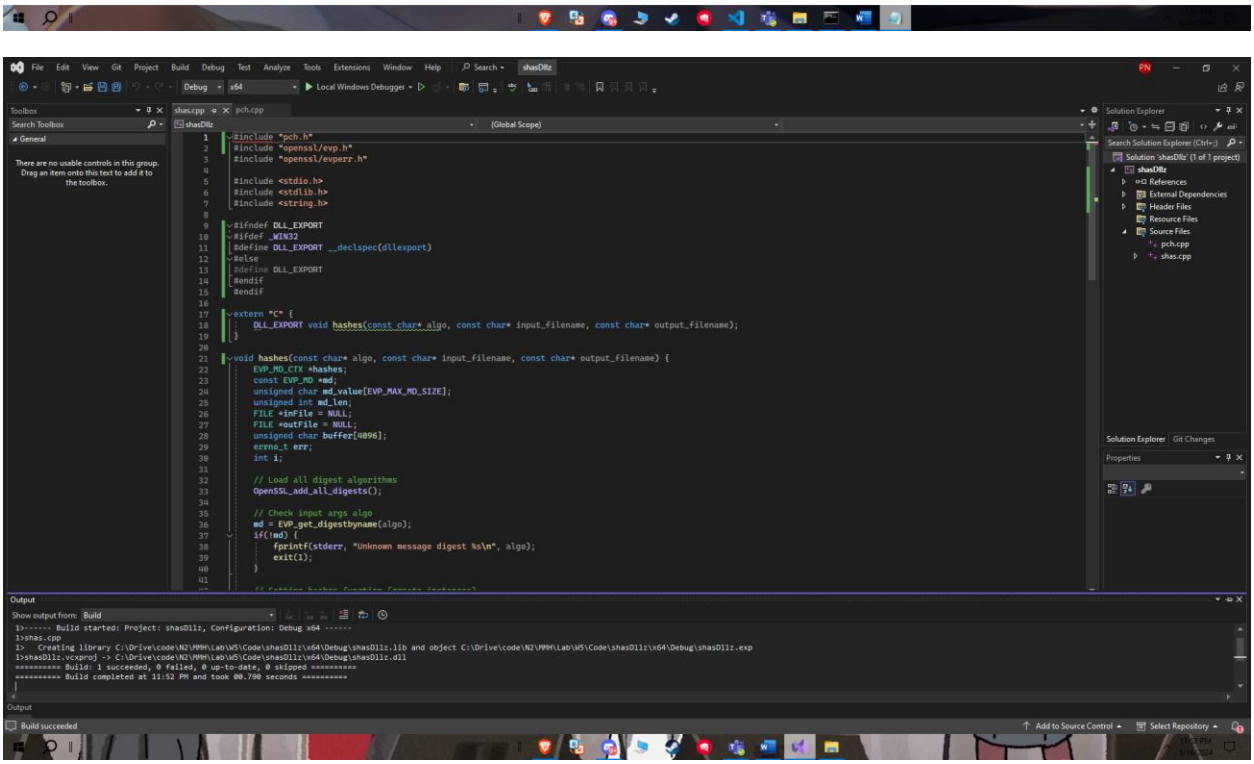
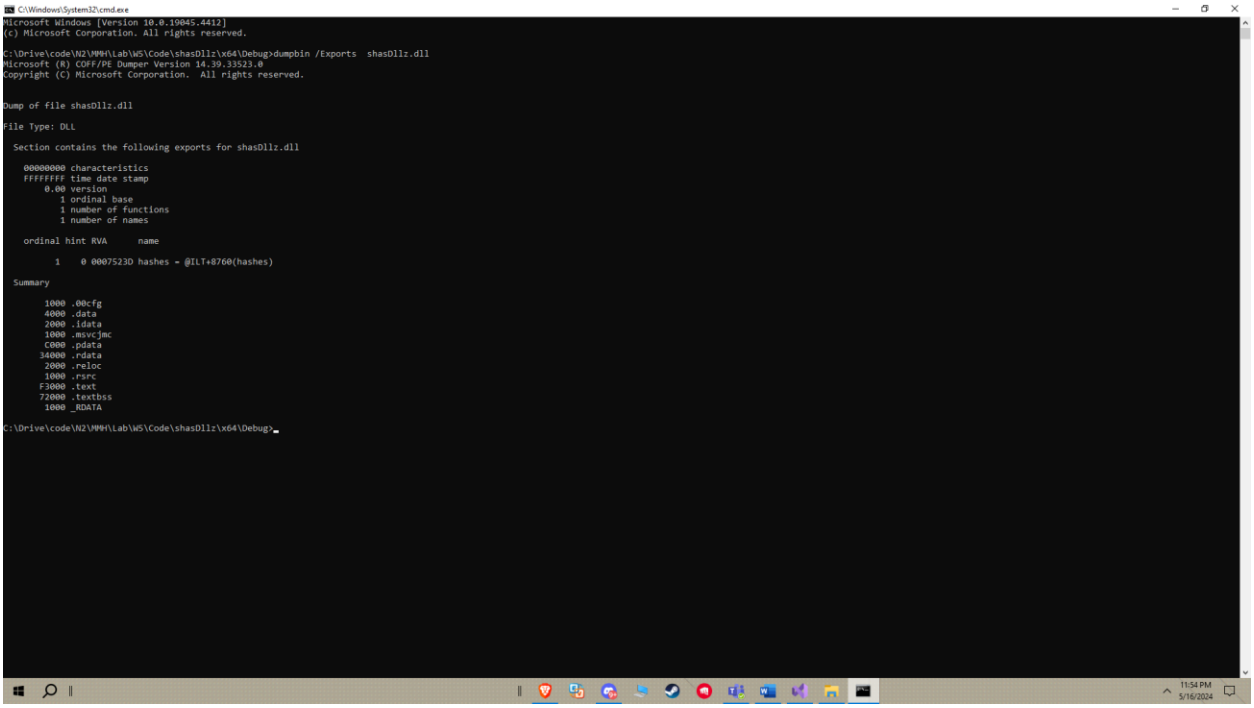
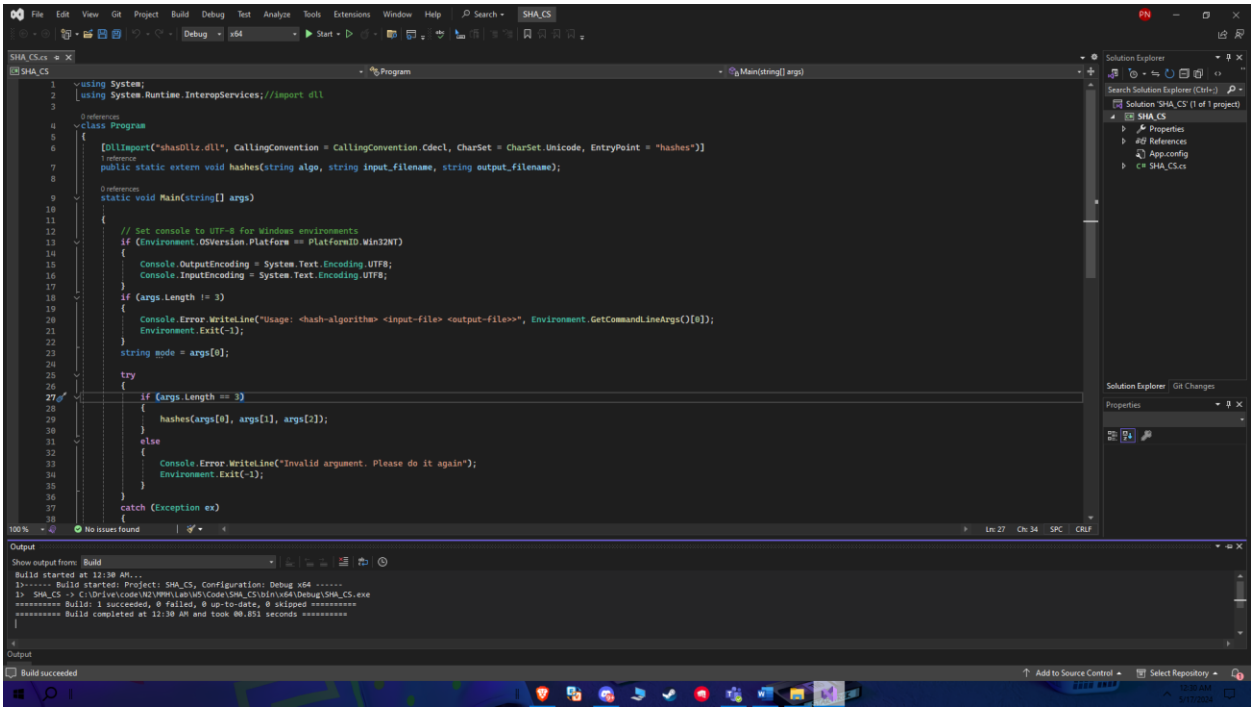

shas.c
#include "openssl/evp.h"
#include "openssl/evperr.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#ifdef BUILD_DLL
#ifdef _WIN32
#define EXPORT __declspec(dllexport)
#else
#define EXPORT __attribute__((visibility("default")))
#endif
#else
#define EXPORT
#endif
EXPORT void hashes(const char* algo, const char* input_filename, const char* output_filename);
EXPORT void hashes(const char* algo, const char* input_filename, const char* output_filename) {
EVP_MD_CTX *hashes;
const EVP_MD *md;
unsigned char md_value[EVP_MAX_MD_SIZE];
unsigned int md_len;
FILE *inFile = NULL;
FILE *outFile = NULL;
unsigned char buffer[4096];
errno_t err;
int i;
// Load all digest algorithms
OpenSSL_add_all_digests();
// Check input args algo
md = EVP_get_digestbyname(algo);
if(!md) {
fprintf(stderr, "Unknown message digest %s\n", algo);
exit(1);
Executing task: cl.exe /MT /GS /W4 /nologo /EHsc /FeC:\Drive\code\N2\MMH\Lab\W5\Code\shas.dll /D BUILD_DLL /IC:\Drive\code\N2\MMH\Lab\W5\Code\include C:\Drive\code\N2\MMH\Lab\W5\Code\shas.c /li
nk C:\Drive\code\N2\MMH\Lab\W5\Code\lib\libcrypto.lib crypt32.lib ws2_32.lib /DLL
shas.c
Creating library C:\Drive\code\N2\MMH\Lab\W5\Code\shas.lib and object C:\Drive\code\N2\MMH\Lab\W5\Code\shas.exp
Terminal will be reused by tasks, press any key to close it.

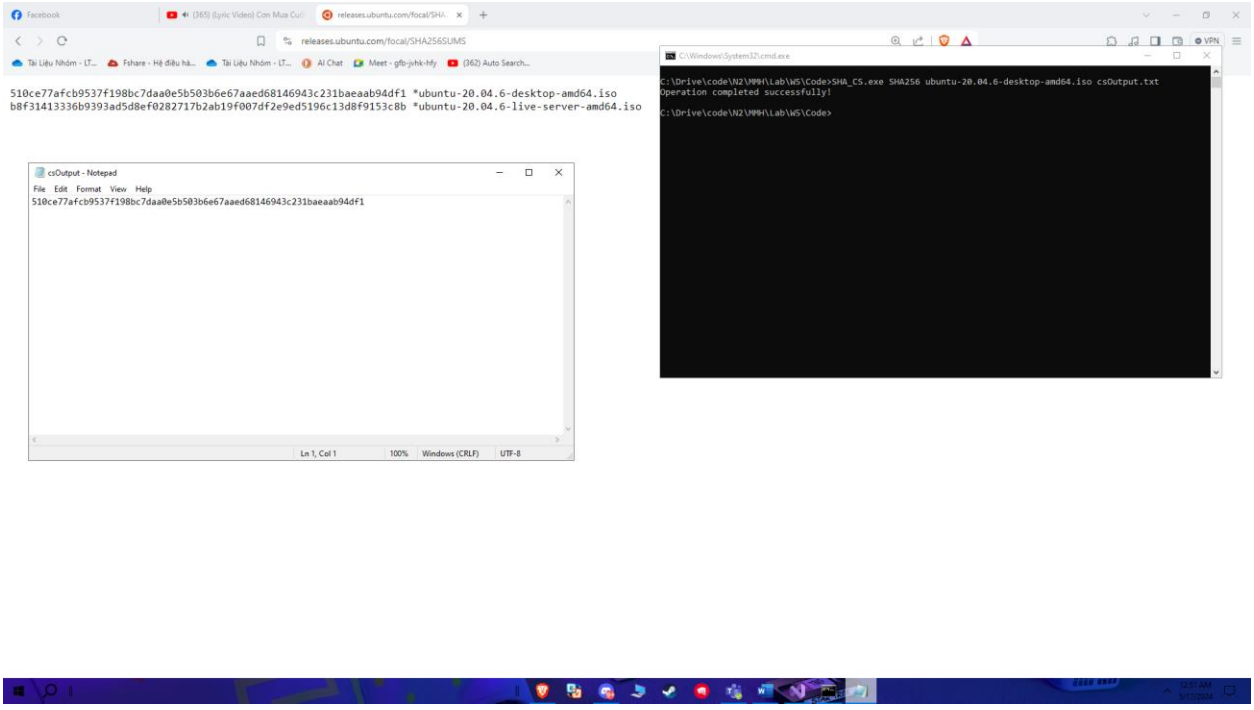

shas.py
import ctypes
import os
from ctypes import c_char_p
import sys
dllpath=os.path.join(os.getcwd(),"shas.dll")
lib = ctypes.CDLL(dllpath, winmode=ctypes.DEFAULT_MODE)
# Set up the prototype of the function
# All of them are strings (char*)
hashes = lib.hashes # call hashes funtion from shas.so;
hashes.argtypes = [c_char_p, c_char_p, c_char_p]
hashes.restype = None # The function returns void
# Wrapped funtions
def call_hashes(algo, input_filename, output_filename):
# Convert Python strings to bytes, as ctypes works with bytes
algo = algo.encode('utf-8')
input_filename = input_filename.encode('utf-8')
output_filename = output_filename.encode('utf-8')
# Call the C function
hashes(algo, input_filename, output_filename)
if __name__ == "__main__":
if len(sys.argv) != 4:
print(f"Usage: python {sys.argv[0]} <<algorithm, <input file name>, <output file name>")
sys.exit(1)
algo = sys.argv[1]
input_filename = sys.argv[2]
output_filename = sys.argv[3]
call_hashes(algo, input_filename,output_filename)
libcrypto-3-x64.dll => not found
PS C:\Drive\code\N2\MMH\Lab\W5\Code> ldd shas.dll
ntdll.dll => /c/WINDOWS/SYSTEM32/ntdll.dll (0x7ffa38930000)
KERNEL32.DLL => /c/WINDOWS/System32/KERNEL32.DLL (0x7ffa370e0000)
KERNELBASE.dll => /c/WINDOWS/System32/KERNELBASE.dll (0x7ffa361b0000)
msvcrt.dll => /c/WINDOWS/System32/msvcrt.dll (0x7ffa37500000)
shas.dll => /c/Drive/code/N2/MMH/Lab/W5/Code/shas.dll (0x7ffa2d5a0000)
libcrypto-3-x64.dll => not found
PS C:\Drive\code\N2\MMH\Lab\W5\Code>


releases.ubuntu.com/focal/SHA256SUMS
510ce77afcb9537f198bc7daa0e5b503b6e67aaed68146943c231baeaab94df1 *ubuntu-20.04.6-desktop-amd64.iso
b8f31413336b9393ad5d8ef0282717b2ab19f007df2e9ed5196c13d8f9153c8b *ubuntu-20.04.6-live-server-amd64.i
pythonDll - Notepad
File Edit Format View Help
510ce77afcb9537f198bc7daa0e5b503b6e67aaed68146943c231baeaab94df1
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
C:\Drive\code\N2\MMH\Lab\W5\Code>python shas.py
Usage: python shas.py <algorithm, <input file name>, <output file name>
C:\Drive\code\N2\MMH\Lab\W5\Code>python shas.py SHA256 ubuntu-20.04.6-desktop-amd64.iso pythonDll.txt
C:\Drive\code\N2\MMH\Lab\W5\Code>


shasDllz
shas.cpp
pch.cpp
#include "pch.h"
#include "openssl/evp.h"
#include "openssl/evperr.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#ifndef DLL_EXPORT
#ifdef _WIN32
#define DLL_EXPORT __declspec(dllexport)
#else
#define DLL_EXPORT
#endif
#endif
extern "C" {
DLL_EXPORT void hashes(const char* algo, const char* input_filename, const char* output_filename);
}
void hashes(const char* algo, const char* input_filename, const char* output_filename) {
EVP_MD_CTX *hashes;
const EVP_MD *md;
unsigned char md_value[EVP_MAX_MD_SIZE];
unsigned int md_len;
FILE *inFile = NULL;
FILE *outFile = NULL;
unsigned char buffer[4096];
errno_t err;
int i;
// Load all digest algorithms
OpenSSL_add_all_digests();
// Check input args algo
md = EVP_get_digestbyname(algo);
if(!md) {
fprintf(stderr, "Unknown message digest %s\n", algo);
exit(1);
}
Output
1>------ Build started: Project: shasDllz, Configuration: Debug x64 ------
1>shas.cpp
1> Creating library C:\Drive\code\N2\MMH\Lab\W5\Code\shasDllz\x64\Debug\shasDllz.lib and object C:\Drive\code\N2\MMH\Lab\W5\Code\shasDllz\x64\Debug\shasDllz.exp
1>shasDllz.vcxproj -> C:\Drive\code\N2\MMH\Lab\W5\Code\shasDllz\x64\Debug\shasDllz.dll
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
========== Build completed at 11:52 PM and took 00.790 seconds ==========
Build succeeded


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.
C:\Drive\code\N2\MMH\Lab\W5\Code\shasDllz\x64\Debug>dumpbin /Exports shasDllz.dll
Microsoft (R) COFF/PE Dumper Version 14.39.33523.0
Copyright (C) Microsoft Corporation. All rights reserved.
Dump of file shasDllz.dll
File Type: DLL
Section contains the following exports for shasDllz.dll
00000000 characteristics
FFFFFFFF time date stamp
0.00 version
1 ordinal base
1 number of functions
1 number of names
ordinal hint RVA name
1 0 0007523D hashes = @ILT+8760(hashes)
Summary
1000 .00cfg
4000 .data
2000 .idata
1000 .msvcjmc
C000 .pdata
14000 .rdata
2000 .reloc
1000 .rsrc
F3000 .text
72000 .textbss
1000 _RDATA
C:\Drive\code\N2\MMH\Lab\W5\Code\shasDllz\x64\Debug>
11:54 PM
5/16/2024


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
SHA_CS
Debug
x64
Start
SHA_CS.cs
SHA_CS
Program
Main(string[] args)
using System;
using System.Runtime.InteropServices;//import dll
0 references
class Program
{
[DllImport("shasDllz.dll", CallingConvention = CallingConvention.Cdecl, CharSet = CharSet.Unicode, EntryPoint = "hashes")]
1 reference
public static extern void hashes(string algo, string input_filename, string output_filename);
0 references
static void Main(string[] args)
{
// Set console to UTF-8 for Windows environments
if (Environment.OSVersion.Platform == PlatformID.Win32NT)
{
Console.OutputEncoding = System.Text.Encoding.UTF8;
Console.InputEncoding = System.Text.Encoding.UTF8;
}
if (args.Length != 3)
{
Console.Error.WriteLine("Usage: <hash-algorithm> <input-file> <output-file>>", Environment.GetCommandLineArgs()[0]);
Environment.Exit(-1);
}
string mode = args[0];
try
{
if (args.Length == 3)
{
hashes(args[0], args[1], args[2]);
}
else
{
Console.Error.WriteLine("Invalid argument. Please do it again");
Environment.Exit(-1);
}
}
catch (Exception ex)
{
100 %
No issues found
Ln: 27 Ch: 34 SPC CRLF
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SHA_CS' (1 of 1 project)
SHA_CS
Properties
References
App.config
SHA_CS.cs
Solution Explorer Git Changes
Properties
Output
Show output from: Build
Build started at 12:30 AM...
1>------ Build started: Project: SHA_CS, Configuration: Debug x64 ------
1> SHA_CS -> C:\Drive\code\N2\MMH\Lab\W5\Code\SHA_CS\bin\x64\Debug\SHA_CS.exe
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
========== Build completed at 12:30 AM and took 00.851 seconds ==========
Output
Build succeeded
Add to Source Control
Select Repository


releases.ubuntu.com/focal/SHA256SUMS
510ce77afcb9537f198bc7daa0e5b503b6e67aaed68146943c231baeaab94df1 *ubuntu-20.04.6-desktop-amd64.iso
b8f31413336b9393ad5d8ef0282717b2ab19f007df2e9ed5196c13d8f9153c8b *ubuntu-20.04.6-live-server-amd64.iso
C:\Drive\code\N2\MMH\Lab\WS\Code>SHA_CS.exe SHA256 ubuntu-20.04.6-desktop-amd64.iso csOutput.txt
Operation completed successfully!
C:\Drive\code\N2\MMH\Lab\WS\Code>
csOutput - Notepad
File Edit Format View Help
510ce77afcb9537f198bc7daa0e5b503b6e67aaed68146943c231baeaab94df1
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8