William W.-Y. Hsu

Department of Computer Science and Engineering Department of Environmental Biology and Fisheries Science National Taiwan Ocean University

CONTENTS

7.1	Introd	uction	49									
7.2	Lab Procedures											
	7.2.1	Quick Hide	50									
	7.2.2	WinVisible	50									
	7.2.3	Security Task Manager	51									
	7.2.4	Intervening Memory	51									
	7.2.5	Secret Maryo Chronicles	54									
	7.2.6	Locating Processes	55									
	7.2.7	Scanning and Modifying Memory Contents	55									
	7.2.8	Reading and Identifying Memory Contents	58									
7.3	Lab Q	uestions	59									
7.4	Lab R	eport	60									

7.1 Introduction

In computing, a process is an instance of a computer program that is being executed. It contains the program code and its current activity. Depending on the operating system (OS), a process may be made up of multiple threads of execution that execute instructions concurrently.

A computer program is a passive collection of instructions; a process is the actual execution of those instructions. Several processes may be associated with the same program; for example, opening up several instances of the same program often means more than one process is being executed.

Multitasking is a method to allow multiple processes to share processors (CPUs) and other system resources. Each CPU executes a single task at a time. However, multitasking allows each processor to switch between tasks that are being executed without having to wait for each task to finish. Depending on the operating system implementation, switches could be performed when tasks perform input/output operations when a task indicates that it can be switched, or on hardware interrupts.

A common form of multitasking is time-sharing. Time-sharing is a method to allow

fast response for interactive user applications. In time-sharing systems, context switches are performed rapidly. This makes it seem like multiple processes are being executed simultaneously on the same processor. The execution of multiple processes seemingly simultaneously is called concurrency.

In this laboratory, you will learn how to modify process states and intervene. Using an open source game as an example, we will use Game Master 9 to intervene with the execution of the game process. You will learn how to locate the process you are intervening with, scan, read, and modify memory contents, and some of the architectural specification of the underlying OS and hardware. You will put into practice of numeric conversions from decimal to hexadecimal and vice versa, and understand how little endian architecture works.

7.2 Lab Procedures

7.2.1 Quick Hide

QuickHide allows you to quickly hide all your open processes, files, programs, etc from the Windows 7 taskbar in a click.

And why would you want to do that? Maybe you are doing something you shouldn't be doing and your dad or Boss comes in. QuickHide could be a life saver — by letting you hide all in a click!

The app QuickHide.exe runs in the background. You can see it in the Task manager, but will not be displayed in the notification area.

Quick hide can be downloaded from

http://www.thewindowsclub.com/hide-taskbar-processes-and-applications-quickly-with-quickhide.



7.2.2 WinVisible

In any company or organizations, there are rules for things to work properly and the truth is that not all the same suit, to avoid having personal distractions, many managers prohibit the execution of instant messaging programs, or P2P programs (such as bitcomet, any instance): What can we do about this situation?

The beauty of this program is that it allows customization sublime as you can give or assign a keyboard shortcut (known as hotkey) to the application you want, examples are many. WinVisible allows you to hide the Windows Live Messenger process, simply by pressing a series of keys together. So fast and simple. All this can be configured.

It is recommended that before you hide any application to save all the information you' re editing at the same it could be lost. WinVisible is a very interesting application.



WinVisible can be downloaded from http://www.neptunecentury.com/projects/winvisible.

7.2.3 Security Task Manager

On your Windows system, you can open your task manager and see how many processes are running. However, there are much more to this. Many of the processes are in fact, hidden, not shown in the task manager.

"Security Task Manager tells you exactly what programs are running on your computer - and it gives you answers to the obvious ensuing questions, such as where these programs reside, who makes them, what they are called, whether they include hidden components, and what all this means to your computer."

Security Task Manager shows all active processes on your computer. You can easily recognize the endangering potential of each process. No other Task Manager or Process Viewer has this feature. Furthermore, you can put a process into quarantine or search the internet for information about that process.

Security Task Manager <u>F</u> ile <u>E</u> dit <u>V</u> iew <u>H</u> elp						
🎊 Windows processes 🔀 Remove	🔀 Quarantine	e 🜔 Google				
Name Rating	File		Title, Description	Manufa		
ctagent 66	C:\Windows\Sy	stem32\ctagent.dll	COV/U-lana Class (Perm	Creative		
Intel(R) Matrix Storage E 42	C:\Program Files	s Java yre 1.6.0_02 olin \ssv.dii s \Intel \Intel Ma\laantmon.exe	RAID Monitor	Intel Con		
C Adobe Reader 8.1.3 - D 32	C:\Program Files	Common F\AcroIEHelper.dll	Adobe PDF Helper for I	Adobe S		
HD Audio Control Panel 24	C:\Program File: C:\Windows\Rt	s/ATT Technologies\CCC.exe HDVCpl.exe	GDI+ Window	Realtek		
CtHelper Application 24	C:\Windows\Sy	stem32\CTHELPER.EXE	CtHelper32	Creative		
CTXfiHlp MFC Application	C:\Windows\Sy C:\Windows\sy	system 32\CTXFIHLP.EXE MIMDEVAPI Device Wi system 32\Ati2evxx.exe ATI External Event Utilit				
🗳 Event Monitor User Notifi	C:\Program Files	s\Intel\Intel Matr\IAAnotif.exe	IAAMonitor Notify App,	Intel Con		
•	m					
Ctagent		Properties	Rat	ing		
Manufacturer Creative Technology Lt Type DLL, hidden Start File C:\Windows\System32 Comment cOlida bers to absorbed	d : ctagent	Could manipulate other progr Window not visible No Windows system file No detailed description availa functions: record inputs, not d	ams	•		
Comment Color here to change t	no maung?	Rating: potentially dangerous				

Security Task Manager can be downloaded from http://www.neuber.com/taskmanager/taskmanager.html.

7.2.4 Intervening Memory

52

This section will explain the mechanics of a program running on an Intel processor. Recall that Intel processors allocate memory using the little endian method, i.e., most significant byte is at the lower address. Consider the following code implementing a simple number guessing game. Using the tool **Cheat Engine v6.8.1** to lock into the process memory, try to examine the memory to win the game in 1 guess.

```
#include <stdio.h>
 1
  #include <stdlib.h>
  #include <time.h>
3
5
   int main( void )
   {
      int answer;
7
      int guess;
9
      srand( time( NULL ) );
      \underline{\mathbf{while}}(1)
11
      {
          answer = (\underline{int})(rand());
13
          guess = -1;
15
17
          <u>while</u>( guess != answer )
          ł
              printf( "Input your guess): " );
19
              scanf("%d", &guess );
21
              \underline{if}( guess < answer )
                 printf( " Too small!!\n");
23
              <u>else</u> <u>if</u>( guess > answer )
                 printf( " Too big!!\n" );
25
              else
                 printf( "Correct! Generating new number.\n");
27
          }
29
      }
31
      return 0;
```

Using your favorite compiler, compile the following code and run the code. In our demonstration, the code is name as *NumberGuessing.exe*, so we lock into the memory of this code using Cheat Engine.

C:\Users\William Hsu\Qsync\Course\Introduction to Computers\CourseMaterials\Labs\New\Processes\Processes\Num	berGuessing.exe — 🗆 🗙
nput your guess:	×
Address Value Fre	Fist scan Vext scan Ondo scan and a
🐔 Process List	×
File	
Applications Processes Windows	
00004534-7) (2) 000034F8- (2) 00003268-Cheat Engine 6.8.1 (7) 0000238C-iTunes (7) 00004550- (7) 00001704- (7) 000001704- (7) 00001704- (7) 00001704- (7) 000017	Asu\Qsync\Course\Introduction to Computers\CourseMaterials\Labs\New Pro urse\Introduction to Computers\CourseMaterials\Labs\New\Processes\Proc
	Open Cancel Attach debugger to process
	Network

We assume the program stores the solution of this guessing game in some address near our input, so let us try to find this address. Our first attempt using 50 as our guess shows 2 memory locations containing this value. This means that we need more effort to trace down our objective.

C:\Users\William Hsu\Qsyr	ic\Course\Introduction to Computers\CourseMat	Cheat Engine 6	ö.8.1			- 0	×
Input your guess: 50 Too small!! Input your guess:	Fi	le Edit Table	D3D H	lelp	000027A0-NumberGuessing.exe		S
	Fou	und: 2					
	A	ddress	Value	Pre	New Scan Next Scan	Undo Scan	(international states)
	0	062FE48	50	50	Value:		Settings
	0	0772BC4	50	50	Hex 🗌 50		
					Scan Type Exact Value	✓ □ Not	
					Value Type 4 Bytes	\sim	

With luck, our second scan using 100 as our input shows that there is only 1 address (0x0062FE48) that follows our input trail. For a 4 byte integer, the 3 positions following 0x0062FE48 (0x0062FE48 to 0x0062FE4B) forms the data. Reading the data shows 0x64000000 in big endian, but we know that Intel CPUs store them as little endian, so the value should be read in reverse as 0x00000064.

C:\Users\William Hsu\Qsync\Course\Introduction to Computers\CourseMat	🐔 Cheat E	ingine 6	.8.1			- 0	×
Input your guess: 50 Too small!! Input your guess: 100	File Edit	Table	D3D	Help	000027A0-NumberGuessing.exe		Ĉ
Too small!! Input your guess:	Found: 1 Address		Valu	e Pre	New Scan Next Scan	Undo Scan	S
	0062FE48		100	100	Value: Hex 🔲 100		Settings
					Scan Type Exact Value	∨ □Not	

Let us look into this memory location. A bold assumption made is that the solution should be stored near our input, and for most Intel compilers, the memory allocated is in reverse order of the computer code. So the next 4 bytes contains our solution 0x62FE4C to 0x62FE4F. This 4 bytes (in little endian) represents the value 0x00005D07 which is 23815, so we input this value as our next guess.

54

File Search View Debug Tools Kernel tools NumberGuessing.exe NumberGuessing.exe File Edit Table D3D NumberGuessing.exe 1500 488385 28 Stopped NumberGuessing.exe+1500 488385 5525F000 mov rax<[NumberGuessing.exe+4460] NumberGuessing.exe+1504 488805 5525F0000000 mov rax<[NumberGuessing.exe+4460] Pressent 10 Address Value Pre	🗧 Memory Viewer		- 🗆 ×	🐔 Cheat Engine 6.8.1			- 0	×
NumberGuessing.exe+1500 000027A0-NumberGuessing.exe Address Bytes Opcode NumberGuessing.exe+1500 48.83 EC 28 sub rsp.28 NumberGuessing.exe+1500 48.88 D5 S52P0000 mov rax, [NumberGuessing.exe+4460] NumberGuessing.exe+1508 C7 00 00000000 mov rax, [NumberGuessing.exe+4460] 10	File Search View Debug Tools	Kernel tools		File Edit Table D30	DН	lelp		
Address Bytes Opcode NumberGuessing.exe+1500 48.83 EC 28 sub rag.28 NumberGuessing.exe+1504 48.88 05 552f0000 mov rag.[NumberGuessing.exe+4460] Perc New Scan Next Scan Undo Scan NumberGuessing.exe+1508 C7 00 00000000 mov rag.[NumberGuessing.exe+4460] 10 Add ress to the addresses to the addressist		NumberGuessing.exe+1	1500	000027A0-NumberGuessing.exe				\sim
NumberGuessing.exe+1500 488.85 C2.8 sub reg/2.0 NumberGuessing.exe+1504 488.85 5525000 mov rat/(NumberGuessing.exe+4460) New Scan Value Pre New Scan Undo Scan NumberGuessing.exe+1504 48.88 05 55250000 mov rat/(NumberGuessing.exe+4460) New Scan Value Pre New Scan Undo Scan Value New Scan New Scan Value New Scan New Scan New Scan New Scan New Scan New Scan	Address Byt	tes Opco	ode ^	Found: 1				\sim
NumberGuessing.exe+1504 48 88 05 552F0000 mov rax,(NumberGuessing.exe+4460) NumberGuessing.exe+1508 C7 00 00000000 mov [rax],00000000 [rax],0000000 dt add esses to the addressist	NumberGuessing.exe+1500 48	83 EC 28 sub	rsp,28	Address Va	lue	Pre New Scan Next Scan	Undo Scan	\sim
NumberGuessing.exe+1508 C7 00 00000000 mov [rax],00000000 Add selected addresses to the add	NumberGuessing.exe+1504 48	8B 05 552F0000 mov	rax,[NumberGuessing.exe+4460]	0062FE48 10				
Number Consider and AEAA ED AAOCOOOD and Number Consider and OACO	NumberGuessing.exe+150B C/	440C0000	[rax],0000000			Add selected addresses to the addressist		
NumberGuessing.exe+1511 Ed 4A0CU000 Call NumberGuessing.exe+2160 Change value of selected addresses Ctrl+E	NumberGuessing.exe+1511 E8	4AUCUUUU call	NumberGuessing.exe+2160			Change value of selected addresses		Ctrl+E
NumberGuessing.exe+1516 E8 95FCFFFF call NumberGuessing.exe+1180 Change value of selected addresses back to previous/saved value Ctrl+Alt+E	NumberGuessing.exe+1516 E8	95FCFFFF call	NumberGuessing.exe+11B0			Change value of selected addresses back to previous/saved	d value 🛛 Ct	rl+Alt+E
NumberGuessing.exe+1518 90 nop Browse this memory region Ctrl+B	NumberGuessing.exe+151B 90	nop				Browse this memory region		Ctrl+B
NumberGuessing.exe+151C 90 nop	NumberGuessing.exe+151C 90	nop			_	Disassemble this memory region		Ctrl+D
NumberGuessing.exe+151D 48 83 C4 28 add rsp.28	NumberGuessing.exe+151D 48	83 C4 28 add	rsp,28					CLUD I
NumberGuessing.exe+1521 C3 ret	NumberGuessing.exe+1521 C3	ret				Remove selected address		Ctri+Dei
NumberGuessing.exe+1522 90 nop Show previous value column	NumberGuessing.exe+1522 90	nop			\checkmark	Show previous value column		
NumberGuessing.exe+1523 90 nop	NumberGuessing.exe+1523 90	nop				n later in the second second		
NimberGuession eve+152/ 90 non Keload the previous value list (Forgot value scan)	NumberGuessing eve+152/ 90	non				Reload the previous value list (Forgot value scan)		
subtract Find out what accesses this address Ctrl+F5		subtract				Find out what accesses this address		Ctrl+F5
Protect: Pard /Write_011ccst(opBarese00430000 Barese00430000 Street000	Protect Pead/Write Allocat	ionBagam00430000 Ba	se=0062E000_Stge=1000			Find out what writes to this address		Ctrl+F6
address 48 49 44 4E 4C 4D 4E 4F 50 51 52 53 54 55 56 57 09ABCDEF01234567	address 48 49 4A 4B 4C 4D	4E 4F 50 51 52 53 5	4 55 56 57 89ABCDEF01234567			(Level de stard)		
0062FE48 64 00 00 00 75 D0 00 D0 13 C0 00 00 00 0]	0062FE48 64 00 00 00 07 5D	00 00 D0 13 C0 00 0	0 00 00 00 d]		_	nexadecimai		
0062FE58 E8 13 40 00 00 00 00 00 00 00 00 00 00 00 00	0062FE58 E8 13 40 00 00 00	00 00 00 00 00 00 0	0 00 00 00 .0	Memory view	•	Default		
0062FE68 7C 00 00 00 00 00 00 02 7A 40 00 00 00 00 0 z@ Byte	0062FE68 7C 00 00 00 00 00	00 00 20 7A 40 00 0	0 00 00 00 z@			Byte		
0062FE78 00 00 00 00 00 00 00 00 00 00 00 00 00	0062FE78 00 00 00 00 00 00	00 00 00 00 00 00 0	0 00 00 00	Active Description		2 Bytes		
	0062FE88 00 00 00 00 00 00	00 00 00 00 00 00 00 0	00 00 00 00					

With the address of the solution locked, we can always guess the number in 1 try.

C:\Users\William Hsu\Osync\Course\Introduction to Computers\CourseMaterials\Labs\New\Processes\Processes\NumberGuessing.exe

C. Cosers (winitam risd (Qsync (Course (introduction to Computers (Course waterials (Labs (New (Processes (Processes (Number Quessing, exe
input your guess: 50
Too small!!
input your guess: 100
Too small!!
input your guess: 23815
orrect! Generating new number.
input your guess:

The address value for the next iteration is 0x00002350 (in little endian), which is 20515 in decimals. Thus our next guess with this value is accepted by the program.

🐔 Mei	moi	y Vi	ewei	r															-		×	
File Se	earc	h '	View	v D	ebu	g 1	Fool	s K	erne	el too	ols											
								N	umb	berG	ues	sing	.exe	+15	00							
Address	s						By	tes					Op	cod	е							^
Number	rGu	essi	ng.e	exe+	150	0	48	83	EC 2	28			sub)	rs	p ,28	}					
Number	NumberGuessing.exe+1504 48							8B 05 552F0000 mov			v	rax,[NumberGuessing.exe+4460]				0]						
Number	NumberGuessing.exe+150B C7 0						00	000	000	00		mo	v	[ra	ax],C	000	0000					
NumberGuessing.exe+1511 E8 4A0C0000 call NumberGuessing.exe+2160																						
NumberGuessing.exe+1516 E8 95FCFFFF call NumberGuessing.exe+11								+11B0														
NumberGuessing.exe+151B 90 nop																						
NumberGuessing.exe+151C 90 nop																						
Number	rGu	essi	ng.e	exe+	151	D	48	83	C4 2	28			ado	ł	rs	p,28	3					
Number	rGu	essi	ng.e	exe+	152	1	C3						ret									
Number	rGu	essi	ng.e	exe+	152	2	90						no	D								
Number	rGu	essi	ng.e	exe+	152	3	90						no	0								
Number	rGu	acci	nae	vo+	152	Λ	٩n							n								
<																					> '	Y
										5	ubt	rac	t									_
Protec	t:I	Read	d/W:	rit	e 1	A110	ocat	io	nBa	se=	004	300	00 1	Base	==0(0621	1000) Siz	e=10	00		^
addres	s	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	89AE	BCDEF	0123456	7	
0062FE	48	FF	FF	FF	FF	23	50	00	00	D0	13	C0	00	00	00	00	00		# ₽			
0062FE	58	E8	13	40	00	00	00	00	00	00	00	00	00	00	00	00	00	.@.	• • • •		•	
0062FF	68	70	00	00	00	00	00	00	00	20	72	40	00	00	00	00	00			714		
	Input your guess: 20515 Correct! Generating new number. Input your gness:																					

7.2.5 Secret Maryo Chronicles

Secret Maryo Chronicles (SMC) is an Open Source two-dimensional platform game with a design similar to classic computer games. SMC has computer support to a great degree by using an accelerated Open Graphics Library for the best possible graphic design and stock performance. Besides introducing open source software which you can use for your studies, let us put Cheat Engine into real action.



Please download the Windows Installer version of the game and install it to your PC. It can be downloaded from http://www.secretmaryo.org/. The latest version is 1.9.

7.2.6 Locating Processes

Cheat Engine is capable of scanning process memory as well as files. Execute SMC first and then use the refresh button on top of the screen to update the system state to GM9 (Use Ctrl+ESC to switch process). Select the process of SMC.

🐔 Cheat Engin	te 6.8.1			-	
File Edit Tab	ole D3D Help	00000740 N			-
🔣 😅 🖬 –		000027A0-INUM	iberGuessing.exe		
Found: 1					- Paph:
Address	Value Pre	New Sca	n Next Scan	Undo S	can 🚬 ur s
🐔 Process List					X
File					
Applications Processe	s Windows				
00003208-Cheat En 00003208-Chures 00004550- 00001704- 00001704- 00001708-Vintekt 1 00000178-XnView C 00000022-Processer 0000022-CProcester 0000045C- 0000045C-	gine o.s. 1 0.3 - [C:\Users\William ps - Inkscape Ilassic - [Clipboard-15] s.pdf - Adobe Acrobat aryo Chronicles	Hsu\Qsync\Cours Pro	e\Introduction to Com	puters\CourseMate	rials\Labs\New
		Open Attach debugs Netw	Cancel ger to process work		

Now you are ready to start scanning and modifying the contents of your running game application.

7.2.7 Scanning and Modifying Memory Contents

We will show you how to lock on to a specific memory position and modify (or lock) is value. Let us use the number of "lives" of your character (which is 3 in the beginning of the game) as an example. We input the value "3" into the search target and press scan. You should see something like the below figure and following of many addresses with the value "3".



After the scan has been completed, return to SMC and commit suicide. The life value of your character should then be deducted by 1, resulting in 2. Use this value as the next scan value. You may observe that after the second scan, the number of results (address values) fulfilling the criteria decreases. Commit suicide again and use the value "1" to scan the third time. A sample result after the second and third scan is shown below.



You can see that only 3 address spaces fulfil the scan criteria, and you can suppose that one of the address space store the number of life of your character. The fourth scan results in only 1 address. This is definitely your target.

🐔 Cheat Engine	6.8.1			- 0	×					
File Edit Table	D3D H	elp								
🗟 🖻 目 0000241C-Secret Maryo Chronicles.exe										
Found: 1					5					
Address	Value	Pre	New Scan Next Scan	Undo Scan	Const August					
025F3340	1	1	Value:		Settings					
			Hex 🗌 1							
			Scan Type Exact Value	∽ □Not						
			Value Type 4 Bytes	\sim						

Double click the address value and enter "99" into the value and press Ok. Your lives will now be locked to 99.

3 🛩 E	a 0000241C-Secret Maryo Chronicles.exe												
Found: 1													
Address		Value	Pre	New Scan N	lext Scan	Undo Scan	(interest						
025F33*0		00					Setting						
	Add se	elected a	ddresses t	H									
	Chang	e value o	f selected a	Ctrl+E									

58	Introduction	n to Computer	rs— La	boratory	manual and exe	ercises
Tile Edit Table	The trained to be a constrained of the trained of trained of the trained of trained o		egion nory region	> Ctrl+B Ctrl+D	Description Address Type Value Undo last edit Smart edit address(es)	Ctrl+Enter Ctrl+Alt+Enter Alt+Enter Enter Ctrl+Z Shift+Ctrl+Alt+Enter
	Active Description	Address 025F3340	Type 4 Bytes	Value 99		

Try suicide again with your character and see what happens.



7.2.8 Reading and Identifying Memory Contents

For most computer programs, we like to put variables together whether in a structure, a class, or declared together. With this assumption in mind, we can try to identify some other information to changes. SMC has other attributes, such as the number of coins collected and current score. Suppose after the previous section, the game stats is shown below:

🐔 Memory Viewer								
File Search View Debug Tools Kernel tools								
				Secr	et Maryo Chronicles.ex	xe+1500		
Address Bytes	Opcode	e	Comment					
Secret Maryo Chronicles.exe+154D	dec	ebp						
Secret Maryo Chronicles.exe+15E4 83	in	al,-7D	131					
Secret Maryo Chronicles.exe+157D E4	jnl	"Secret Maryo Chronicles.exe"+14	4ES->Secret Maryo					
Secret Maryo Chronicles.exe+1500 7E 7A	add	[esi+7A],bh						
Secret Maryo Chronicles.exe+158B 55 08	mov	edx,[ebp+08]						
Secret Maryo Chronicles.exe+158B 02	mov	eax.[edx]						
Secret Marvo Chronicles.exe+158B 4D 08	mov	ecx.[ebp+08]						
Secret Marvo Chronicles exe+1503 48 04	add	ecx [eax+04]						
Secret Maryo Chronicles exe+15EE 15 18055000	call	dword ptr ["Secret Marvo Chronic	les->MSVCP90 std					
Secret Marvo Chronicles exe+1588 45 BF	mov	[ebp-41] al						
Secret Maryo Chronicles exe+1588 4D 08	mov	erx [ebn+08]						
Secret Maryo Chronicles eve+158B 11	mov	edv [ecv]						
Correct Manyo Chronicles ave 1 15 00 11	-	eux[eux]						
					decrement by 1	1		
Protect:Read/Write AllocationBase=025E0	000 Base	=025F3000 Size=EC000						
address A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A	A AB AC	AD AE AF B0 B1 B2 B3 B4 B5	B6 B7 B8 B9 BA BB	B BC BD BE BF CO CI	C2 C3 C4 C5 C6 0	C7 C8 C9 CA CB CC CI	CE CF D0 D1 D2 D3 D4 D5 1	D6 D7 0123456789ABCDEF0123456789ABCDEF
025F30A0 00 00 00 00 00 00 00 00 00 00 00 0	0 43 00	00 00 00 00 00 00 00 00 00	00 00 00 00 00 00	0 00 00 00 00 00 00	0 00 00 00 00 00 0	00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 0	00 00
025F30D8 00 00 00 00 00 00 00 00 00 00 00 0	0 00 00	00 00 00 00 00 00 00 00 00	00 00 00 00 00 00	0 00 00 00 00 00 00	0 00 00 00 00 00 0	00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 0	00 00
025F3110 00 00 00 00 00 00 00 00 00 00 00 0	0 00 00	00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00	0 00 00 00 00 00 00	0 00 00 00 00 00 00 00 00 00 00 00 00 0	00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 0	00 00
025F3148 00 00 00 00 00 00 00 00 00 00 00 00 00	0 00 00	00 00 00 70 E8 19 77 B0 DE	19 77 00 00 00 00	0 00 00 00 00 00 00				00 00
025F3188 00 00 68 42 00 00 F8 42 33 73 2	C 43 00	00 68 42 00 00 68 42 00 00	02 43 33 73 81 42	3 00 00 F0 41 00 00	0 00 00 C8 2D 11 0	41 00 00 20 41 00 00	E E 42 33 73 AC 43 55 98 (C 3D bB B3s C bB bB C3s C 2
025F31F0 00 00 E8 42 00 00 AB 43 00 00 0	0 00 00	00 00 00 00 00 00 00 00 00	00 00 00 00 00 00	0 00 00 00 00 00 00	0 00 00 00 00 00 0	00 00 00 01 00 01 00	00 00 00 00 80 3F 00 00	80 3F B C
025F3228 00 00 80 3F 00 00 80 3F FF FF F	F FF 00	00 00 00 BD 37 86 35 BD 37	86 35 BD 37 86 35	5 03 00 00 00 07 00	0 00 00 01 00 00 0	00 61 00 00 00 4D 61	. 72 79 6F 00 00 00 63 00 1	00 00 ? ? 7 5 7 5 7 5
025F3260 49 00 00 00 05 00 00 0F 00 0	0 00 67	00 00 00 00 00 00 00 27 00	00 00 20 00 00 00	0 75 00 00 00 00 00	0 00 00 OF 00 00 (00 00 01 00 00 E8 03	00 00 00 00 00 00 00 00 0	00 00 I
025F3298 00 00 00 FF 01 01 00 00 D8 B8 C	6 OD 20	00 00 00 74 00 00 00 00 00	00 00 00 00 00 00	0 00 00 00 00 00 00	0 00 00 00 00 00 00 00 00 00 00 00 00 0	00 00 00 00 00 01 00	00 00 01 00 00 00 01 00 0	00 00 <mark>t</mark>
025F32D0 18 39 8A 05 00 00 00 00 00 00 0	0 00 00	00 00 00 01 00 00 00 00 00	00 00 00 00 00 00	0 00 00 00 00 E8 03	00 00 00 00 00 00 00	00 48 BA C6 0D 20 00	00 00 77 00 00 00 D8 73	65 02 .9
	0 00 00	00 88 41 00 00 80 40 00 00						00 00 te. te
025F3378 00 00 80 3F 26 5C 5F 42 26 5C 5	F 42 53	OE 5D 42 00 00 00 00 00 00	00 00 FF FF FF FF	F 00 00 00 00 00 00 00			00 00 30 77 6D 9D 00 0D	00 80 ?s\ Bs\ BS.1B
025F33B0 32 00 00 00 36 00 00 00 2F 00 0	0 00 31	00 00 00 30 00 00 00 2F 00	00 00 32 00 00 00	0 30 00 00 00 31 00	00 00 38 00 00 0	00 20 00 00 00 32 00	00 00 32 00 00 00 3A 00	00 00 26/10/20
025F33E8 31 00 00 00 39 00 00 00 3A 00 0	0 00 31	00 00 00 37 00 00 00 20 00	00 00 28 00 00 00	0 49 00 00 00 6E 00	0 00 00 66 00 00 0	00 6F 00 00 00 29 00	00 00 20 00 00 00 09 00 0	00 00 19 <mark>:17(I</mark>
025F3420 57 00 00 00 69 00 00 00 6E 00 0	0 00 64	00 00 00 6F 00 00 00 77 00	00 00 20 00 00 00	0 27 00 00 00 43 00	0 00 00 45 00 00 0	00 47 00 00 00 55 00	00 00 49 00 00 00 3A 00 0	00 00 Wi <mark>nd</mark> ow'
025F3458 3A 00 00 00 53 00 00 00 79 00 0	0 00 73	00 00 00 74 00 00 00 65 00	00 00 6D 00 00 00	0 3A 00 00 00 3A 00	0 00 00 64 00 00 0	00 65 00 00 00 66 00	00 00 61 00 00 00 75 00 0	00 00 :System:
025F3490 6C 00 00 00 74 00 00 00 5F 00 0	0 00 55	00 00 00 61 00 00 00 75 00	00 00 74 00 00 00	0 65 00 00 00 55 00		00 6F 00 00 00 6F 00		00 00 1t
025F3500 27 00 00 00 54 00 00 00 61 00 0	0 00 52	00 00 00 61 00 00 00 72 00	00 00 65 00 00 00	0 78 00 00 00 40 00	00 00 6F 00 00 0	00 6F 00 00 00 6B 00	00 00 27 00 00 00 54 00 1	00 00 'Tabarez
025F3538 6F 00 00 00 6F 00 00 00 6C 00 0	0 00 74	00 00 00 69 00 00 00 70 00	00 00 27 00 00 00	0 20 00 00 00 68 00	00 00 61 00 00 0	00 73 00 00 00 20 00	00 00 62 00 00 00 65 00	00 00 oo1t1p
025F3570 65 00 00 00 6E 00 00 00 20 00 0	0 00 63	00 00 00 72 00 00 00 65 00	00 00 61 00 00 00	0 74 00 00 00 65 00	0 00 00 64 00 00 0	00 2E 00 00 00 20 00	00 00 28 00 00 00 30 00 1	00 00 encreat
025F35A8 44 00 00 00 43 00 00 00 42 00 0	0 00 32	00 00 00 45 00 00 00 35 00	00 00 38 00 00 00	0 29 00 00 00 0A 00	0 00 00 00 00 00 0	00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 0	00 00 DC <mark>B</mark> 2 <mark>E</mark> 5 <mark>8</mark>)
025F35E0 00 00 00 00 00 00 00 00 00 00 00 0	0 00 00	00 00 00 00 00 00 00 00 00	00 00 00 00 00 00	0 00 00 00 00 00 00	0 00 00 00 00 00 0	00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 0	00 00
025F3618 00 00 00 00 00 00 00 00 00 00 00 0	0 00 00	00 00 00 00 00 00 00 00 00	00 00 00 00 00 00	0 00 00 00 00 00 00	0 00 00 00 00 00 00 00 00 00 00 00 00 0		00 00 00 00 00 00 00 00 00 0	00 00
			[5]	222 3				

The number of lives is 98 (0x63) (because we set it to 99 in the last section and suicide once to see the result), number of coins is 3, and the current score is 15. Converting the values

to hexadecimal (you should know how to do that!), we have 0x62 (98), 0x03 (3), and 0x0F (15). Recall that Intel hardware are little endian, thus the value of 0x0F will be stored in the memory as 0x00 0x0F. Right click on the memory address where we lock the character life and selected Browse this memory region. We should see the process memory as shown below. This is the exact image of the memory of the current executing program.

Protect:Kead/Write AllocationBase=025E0000 Base=025F3000 Size=EC000																																													
address	A0	Al	A 2	A3	Α4	A 5	A 6	A7	A8	A 9	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	СВ	CC (
025F30A0	00	00	00	00	00	00	00	00	00	00	00	43	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
025F30D8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
025F3110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
025F3148	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	70	E8	19	77	BO	DE	19	77	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
025F3180	75	77	A 8	9D	00	0C	00	9C	3C	OF	52	00	50	EB	98	05	65	00	00	00	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	C8	2D	11	OF	C8	2D	11	0F	00
025F31B8	00	00	68	42	00	00	E8	42	33	73	AC	43	00	00	68	42	00	00	68	42	00	00	02	43	33	73	B1	43	00	00	FO	41	00	00	40	42	00	00	60	41	00	00	20	41	00
025F31F0	00	00	E8	42	00	00	AB	43	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	01	00	01
025F3228	00	00	80	ЗF	00	00	80	ЗF	FF	FF	FF	FF	00	00	00	00	BD	37	86	35	BD	37	86	35	BD	37	86	35	03	00	00	00	07	00	00	00	01	00	00	00	61	00	00	00	4D
025F3260	49	00	00	00	05	00	00	00	OF	00	00	00	67	00	00	00	00	00	00	00	27	00	00	00	20	00	00	00	75	00	00	00	00	00	00	00	OF	00	00	00	00	01	00	00	E8
025F3298	00	00	00	FF	01	01	00	00	D8	B 8	C6	0D	20	00	00	00	74	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	01
025F32D0	18	39	8A	05	00	00	00	00	00	00	00	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	E8	03	00	00	00	00	00	00	48	BA	C6	0D	20
025F3308	88	74	65	02	C8	74	65	02	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
025F3340	63	00	00	00	00	00	00	00	00	01	00	00	00	00	88	41	00	00	80	40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
025F3378	00	00	80	ЗF	74	ЗF	ED	42	74	3F	ED	42	8A	18	EC	42	00	00	00	00	00	00	00	00	FF	FF	FF	FF	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
025F33B0	32	00	00	00	36	00	00	00	2F	00	00	00	31	00	00	00	30	00	00	00	2 F	00	00	00	32	00	00	00	30	00	00	00	31	00	00	00	38	00	00	00	20	00	00	00	32
025F33E8	31	00	00	00	39	00	00	00	3A	00	00	00	31	00	00	00	37	00	00	00	20	00	00	00	28	00	00	00	49	00	00	00	6E	00	00	00	66	0.0	00	0.0	6F	00	00	00	29

We can see memory location close to the life and modify it to 0x4B (75 coins) and 0x38BE07 (3718663 current score). Remember we must change it into little endian representation, i.e., 0x07~0xBE~0x38. (The image below has it addresses different because the program was re-executed)



See what will happen! We have changed the value while the program is running. Of course, if you can not identify the positions of these variables, you can conduct a new search with Cheat Engine.

7.3 Lab Questions

- Show the TA you can hide your taskbar and reveal it. In addition, locate the process of Quick Hide using the task manager in windows and tell the TA how much memory it is using.
- Use WinVisible to hide your media player window while playing a song. (Find a song to play on the network).

You only need to complete 1 of the 2 exercises above.

- Execute Security Task Manager using "Administrator" privileges. How many "extra" process are found compared to the normal windows task manager? Save the process table to a file. How many "Toolbars" applications are running on your computer?
- Show the TA you can always win the guessing game in 1 try (without modifying the source code).
- For the following piece of computer code, it is much harder to lock into the address of the solution. Can you explain why?

```
#include <stdio.h>
   #include <stdlib.h>
 2
   #include <time.h>
   int main( void )
   {
 6
       int *answer;
       int guess;
 8
       srand( time( NULL ) );
10
       \underline{\mathbf{while}}(1)
12
        {
           answer = (\underline{int}^*) malloc(\underline{sizeof}(\underline{int}));
*answer = (\underline{int})(rand());
14
           guess = -1;
16
           \underline{while}( guess != *answer )
18
            {
                printf( "Input your guess: " );
20
                scanf( "%d", &guess );
22
                if( guess < *answer )
printf( " Too small!!\n");</pre>
24
                <u>else</u> <u>if</u>( guess > *answer )
                   printf( " Too big!!\n" );
26
                else
                    printf( "Correct! Generating new number.\n");
^{28}
           }
30
           free( answer );
       }
32
34
       <u>return</u> 0;
```

For the game intervention:

- Lock the number of life of your character to 60 life.
- Lock the number of coins of your character to 90.
- Set your current score to 1234567.
- (BONUS, Harder) Set the current playing time to 0:10.

7.4 Lab Report

Demonstration will be done in class. No late demonstrations will be accepted.