

7

Processes

William W.-Y. Hsu

Department of Computer Science and Engineering

Department of Environmental Biology and Fisheries Science

National Taiwan Ocean University

CONTENTS

7.1	Introduction	49
7.2	Lab Procedures	50
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 Questions	59
7.4	Lab Report	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>.

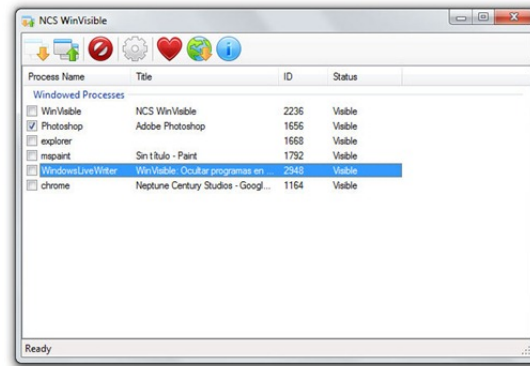


7.2.2 WinVisible

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.



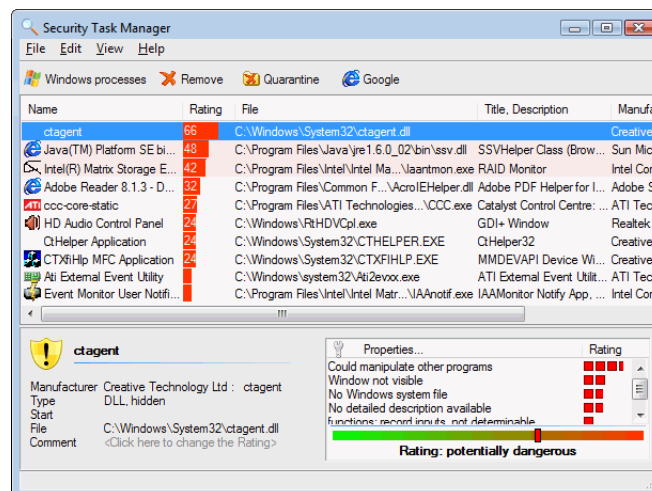
WinVisible can be downloaded from
<http://www.neptuneecentury.com/projects/winvisible>.

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.



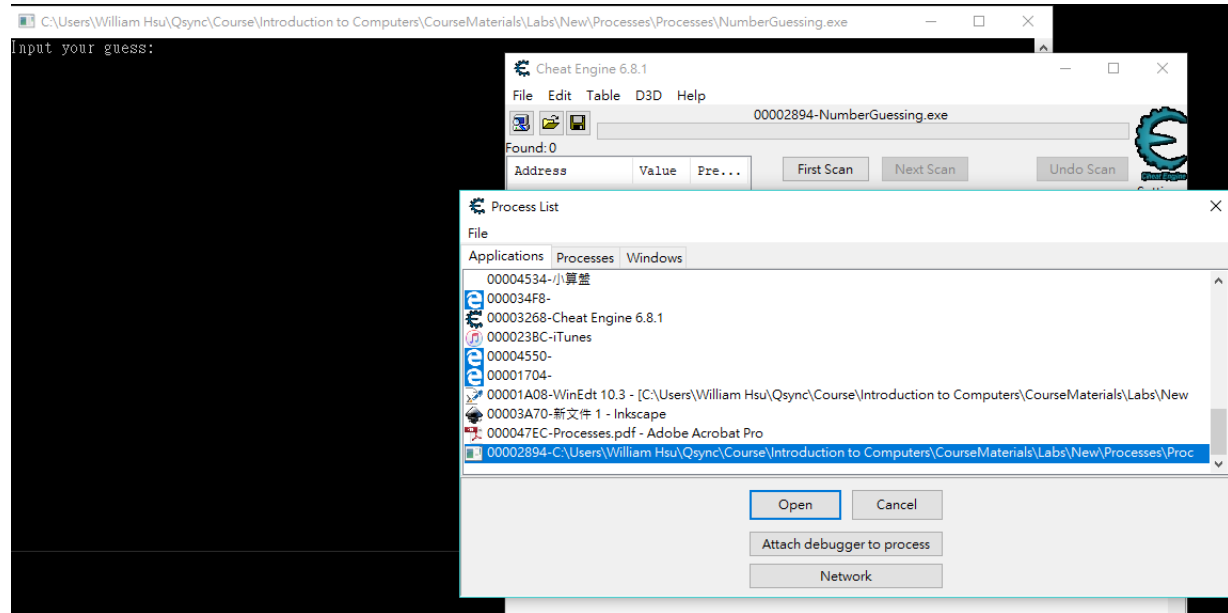
Security Task Manager can be downloaded from
<http://www.neuber.com/taskmanager/taskmanager.html>.

7.2.4 Intervening Memory

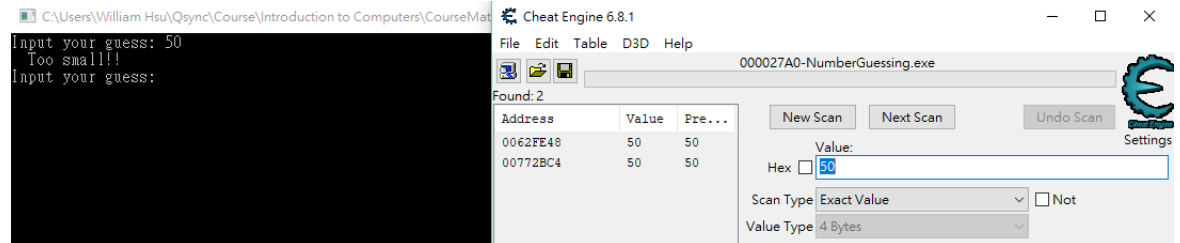
This section will explain the mechanics of a program running on an Intel processor. Recall that Intel processors allocate memory using the little endian method, i.e., most significant byte is at the lower address. Consider the following code implementing a simple number guessing game. Using the tool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

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main( void )
6 {
7     int answer;
8     int guess;
9
10    srand( time( NULL ) );
11    while( 1 )
12    {
13        answer = ( int )( rand() );
14
15        guess = -1;
16
17        while( guess != answer )
18        {
19            printf( "Input your guess): " );
20            scanf( "%d", &guess );
21
22            if( guess < answer )
23                printf( " Too small!!\n" );
24            else if( guess > answer )
25                printf( " Too big!!\n" );
26            else
27                printf( "Correct! Generating new number.\n" );
28        }
29    }
30
31    return 0;
32 }
```

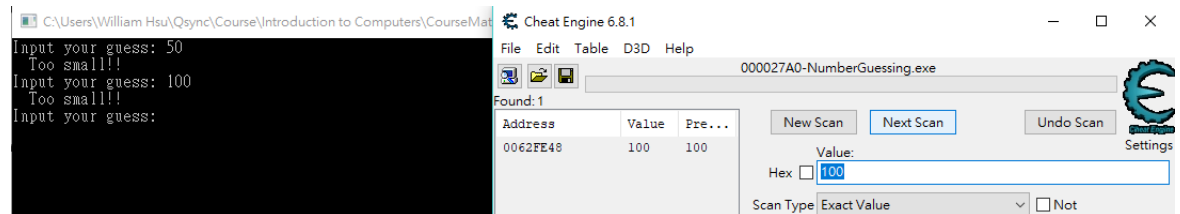
Using your favorite compiler, compile the following code and run the code. In our demonstration, the code is name as *NumberGuessing.exe*, so we lock into the memory of this code using Cheat Engine.



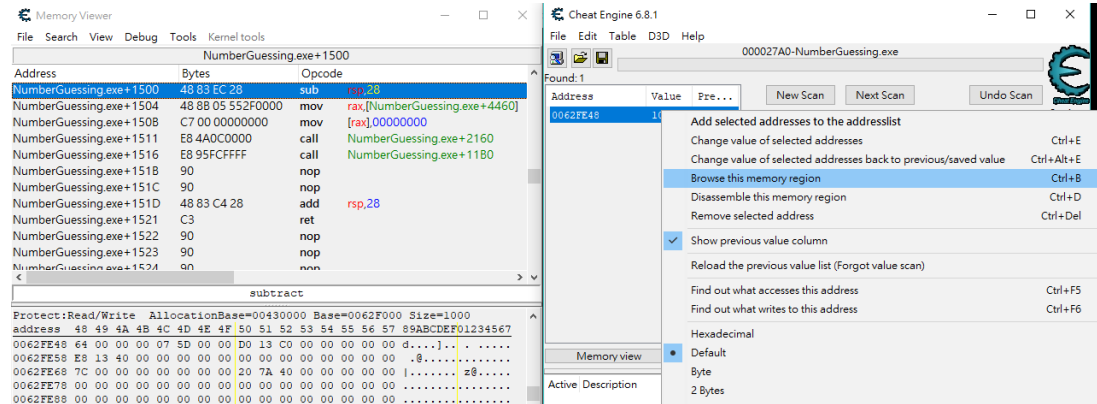
We assume the program stores the solution of this guessing game in some address near our input, so let us try to find this address. Our first attempt using *50* as our guess shows 2 memory locations containing this value. This means that we need more effort to trace down our objective.



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.



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.



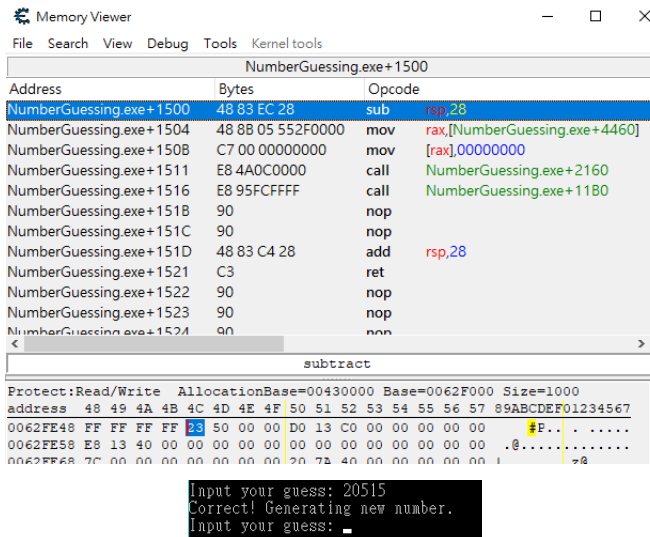
With the address of the solution locked, we can always guess the number in 1 try.

```

C:\Users\William Hsu\Qsync\Course\Introduction to Computers\CourseMaterials\Lab\New\Processes\Processes\NumberGuessing.exe
Input your guess: 50
Too small!!
Input your guess: 100
Too small!!
Input your guess: 23815
Correct! Generating new number.
Input your guess:

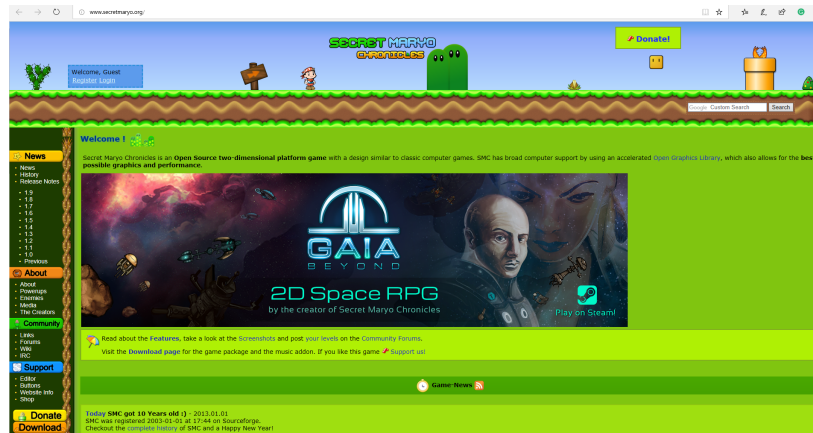
```

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.



7.2.5 Secret Maryo Chronicles

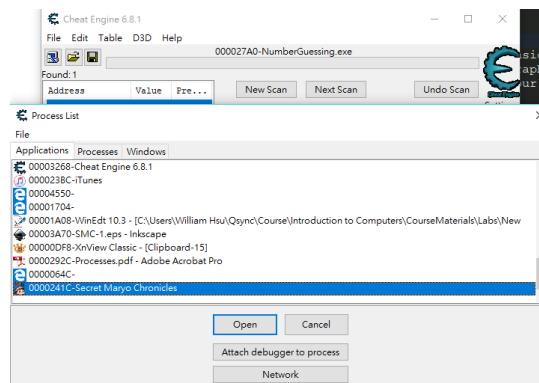
Secret Maryo Chronicles (SMC) is an Open Source two-dimensional platform game with a design similar to classic computer games. SMC has computer support to a great degree by using an accelerated Open Graphics Library for the best possible graphic design and stock performance. Besides introducing open 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

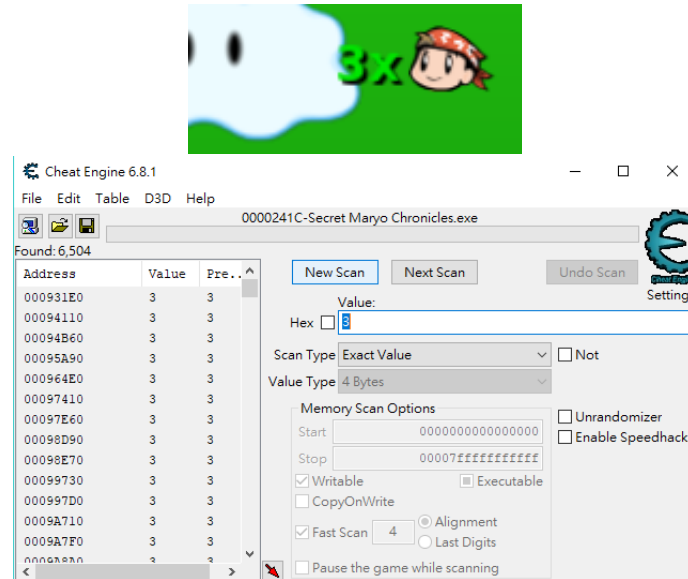
Cheat Engine is capable of scanning process memory as well as files. Execute SMC first and then use the refresh button on top of the screen to update the system state to GM9 (Use Ctrl+ESC to switch process). Select the process of SMC.



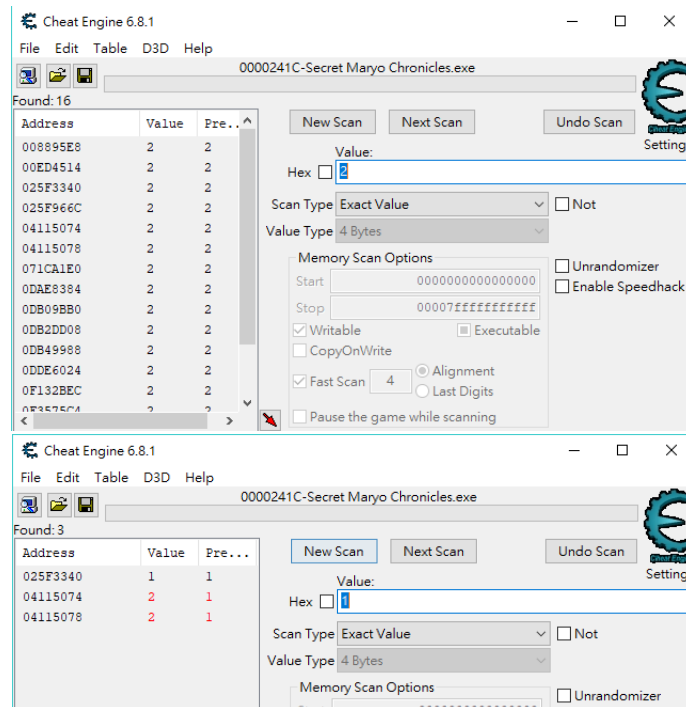
Now you are ready to start scanning and modifying the contents of your running game application.

7.2.7 Scanning and Modifying Memory Contents

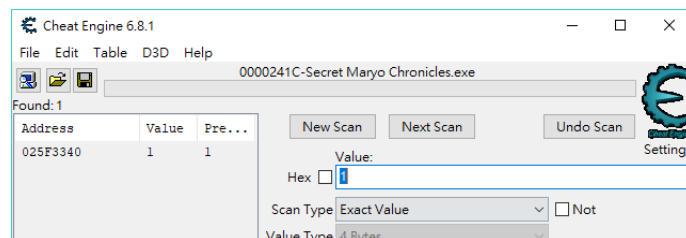
We will show you how to lock on to a specific memory position and modify (or lock) its value. Let us use the number of “lives” of your character (which is 3 in the beginning of the game) as an example. We input the value “3” into the search target and press **scan**. You should see something like the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



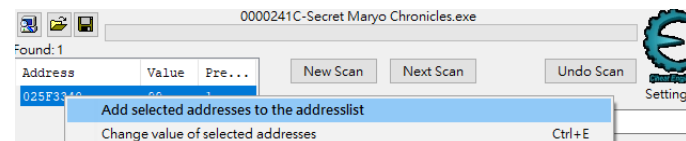
After the scan has been completed, return to SMC and commit suicide. The life value of your character should then be deducted by 1, resulting in 2. Use this value as the next scan value. You may observe that after the second scan, the number of results (address values) fulfilling the criteria decreases.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

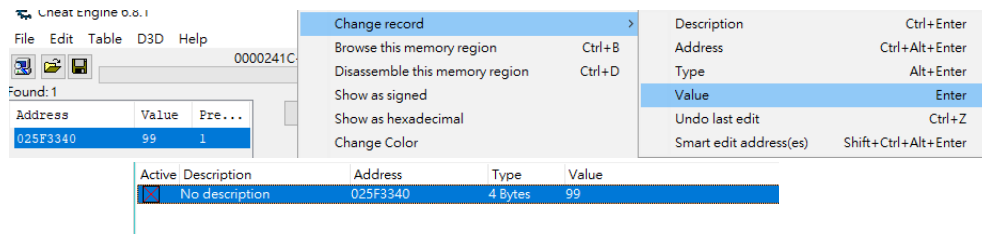


You can see that only 3 address spaces fulfil the scan criteria, and you can suppose that one of the address space store the number of life of your character. The fourth scan results in only 1 address. This is definitely your target.

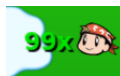


Double click the address value and enter "99" into the value and press Ok. Your lives will now be locked to 99.



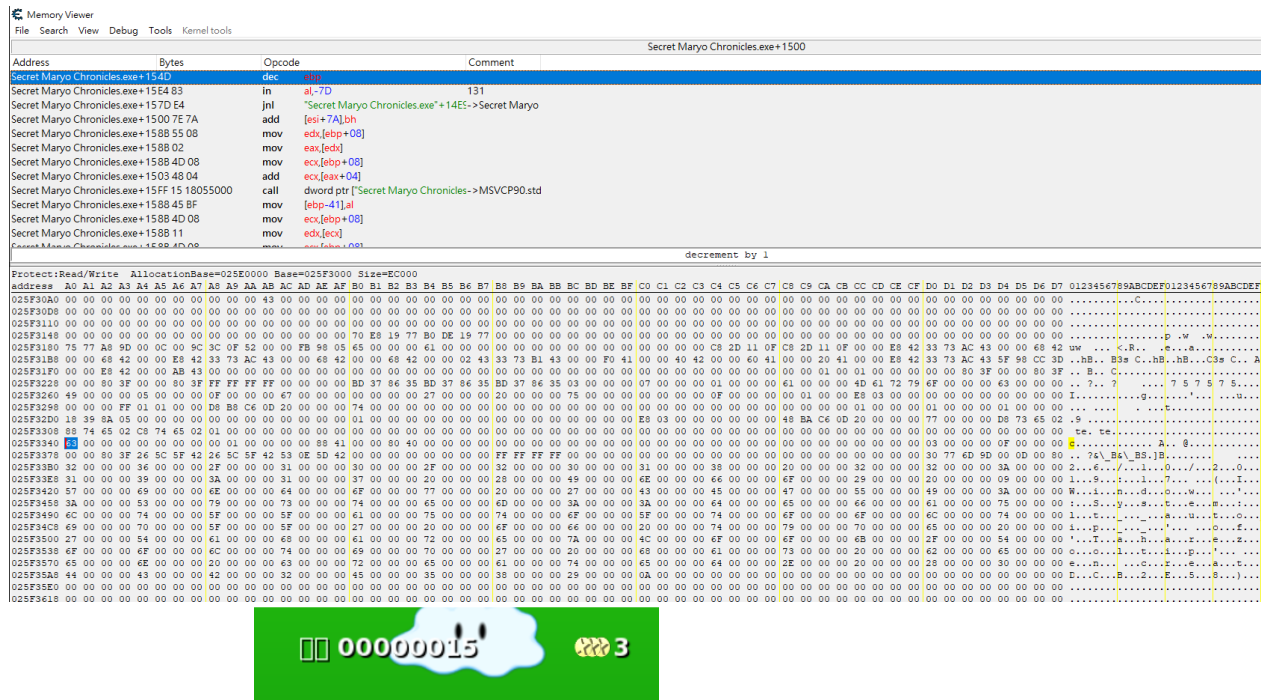


Try suicide again with your character and see what happens.



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:



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

protect:read/write	AllocationBase=025F3000 Base=025F3000 Size=20000																																	
address	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	BB	B9	BA	BB	BC	BD	BE			
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		
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		
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		
025F3148	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	70	E8	19	77	B0	DE	19	77	00	00	00	00	00	00	00	00	00	
025F3180	75	77	A8	9D	0C	00	9C	3C	0F	52	00	50	EB	98	05	65	00	00	00	00	61	00	00	00	00	00	00	00	00	00	00	00	00	
025F31B8	00	68	42	00	00	E8	42	33	73	AC	43	00	00	68	42	00	00	00	68	42	00	00	00	02	43	33	73	B1	43	00	00	00		
025F31F0	00	00	E8	42	00	E8	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		
025F3228	00	00	80	3F	00	00	80	3F	FF	FF	FF	FF	FF	00	00	00	BD	37	86	35	BD	37	86	35	BD	37	86	35	03	00	00	00	00	
025F3260	49	00	00	00	05	00	00	00	0F	00	00	00	67	00	00	00	00	00	00	00	2D	00	00	00	2D	00	00	00	00	75	00	00	00	
025F3298	00	00	00	FF	01	01	00	00	D8	B8	C6	0D	20	00	00	00	74	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
025F32D0	18	39	84	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	00	
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	
025F3340	63	00	00	00	00	00	00	00	00	01	00	00	00	00	00	00	88	41	00	80	40	00	00	00	00	00	00	00	00	00	00	00	00	00
025F3378	00	00	80	3F	74	3F	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	
025F33B0	32	00	00	00	36	00	00	00	2F	00	00	00	31	00	00	00</																		

03718668 76

- 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).

- Execute Security Task Manager using “Administrator” privileges. How many “extra” processes 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?

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main( void )
6  {
7      int *answer;
8      int guess;
9
10     srand( time( NULL ) );
11     while( 1 )
12     {
13         answer = ( int * ) malloc( sizeof( int ) );
14         *answer = ( int )( rand() );
15
16         guess = -1;
17
18         while( guess != *answer )
19         {
20             printf( "Input your guess: " );
21             scanf( "%d", &guess );
22
23             if( guess < *answer )
24                 printf( " Too small!!\n" );
25             else if( guess > *answer )
26                 printf( " Too big!!\n" );
27             else
28                 printf( "Correct! Generating new number.\n" );
29         }
30         free( answer );
31     }
32
33     return 0;
34 }
```

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.