Chapter 4 Part 2

Eric Pereira
CSE3120: Section 01

September 22nd, 2018

1.

`message db "smartest"`
Submit a program that computes the sum of all the byte values in the original message, and stores the result as a number on 4 bytes stored in the last four bytes of the message.

```
.386
.model flat,stdcall
.stack 4096
ExitProcess PROTO, dwExitCode:DWORD
.data
message db "smartest"
target BYTE SIZEOF message DUP(0)
count EQU LENGTHOF message
.code
main PROC
mov esi,0
mov ecx,SIZEOF message
mov eax, 0
mov ebx, 0
lea edx, message
add edx, 4
```

```
L1:
mov al,message[esi]
add ebx, eax
inc esi
loop L1
mov [edx], ebx
invoke ExitProcess,0
main ENDP
END main
```

4.4.5

1. (True/False): Any 32-bit general-purpose register can be used as an indirect operand.

True

5.8

1. Which instruction pushes all of the 32-bit general-purpose registers on the stack?

PUSHAD

12. (True/False): The USES operator only generates PUSH instructions, so you must code POP instructions yourself

False

5.82

1. Write a sequence of statements that use only PUSH and POP instructions to exchange the values in the EAX and EBX registers (or RAX and RBX in 64-bit mode).

```
.386
.model flat,stdcall
.stack 4096
ExitProcess PROTO, dwExitCode:DWORD
.code
main PROC
MOV EAX, 2
MOV EBX, 4
PUSH EAX
PUSH EBX
POP EAX
POP EBX
INVOKE ExitProcess,0
main ENDP
END main
```

2. Suppose you wanted a subroutine to return to an address that was 3 bytes higher in memory than the return address currently on the stack. Write a sequence of instructions that would be inserted just before the subroutine's RET instruction that accomplish this task.

```
.386
.model flat,stdcall
.stack 4096
ExitProcess PROTO, dwExitCode:DWORD
.code
main PROC
CALL subRoutine
```

```
INVOKE ExitProcess,0
main ENDP
subRoutine PROC
ADD EBP, 7
RET
subRoutine ENDP
END main
```