

Assembly Programming (CSE3030-02), Midterm Exam, Spring 2019

09:00-10:15am, April 24 2018 (Wednesday), Instructor: Prof. Youngjae Kim

Student ID: _____ Name: _____

1. Basics of Processor Architecture (23 points)

- 1) [2pt] What is the program that combines object files into an executable program?
- 2) [2pt] Specify true or false on the following statements:
 - a. [1pt] An instruction is executed at assembly time. (True or False)
 - b. [1pt] A directive is executed at assembly time. (True or False)
- 3) [2pt] What is the largest unsigned integer that may be stored in 16 bits?
- 4) [2pt] What is the smallest signed integer that may be stored in 25 bits?
- 5) [4pt] Convert hexadecimal number 0xA2BDF7B5h to octal and binary.
- 6) [4pt] What is the decimal representation of the following signed binary numbers?
 - a. [2pt] 10101101
 - b. [2pt] 11110010
- 7) [4pt] What is the 8-bit binary (two's complement) representation of each of the following signed decimal integers?
 - a. [2pt] -127
 - b. [2pt] -111
- 8) [3pt] Define paging in OS.

2. Simple Assembly Language (26 points)

- 1) [5pt] What are the values of AL register and overflow (OF) and carry flags (CF) after the execution of code below? Justify your answer.

```
mov al, 88h
add al, 90h          ; al= , OF= , CF=
```

- 2) [13pt] Answer the following questions:

```
.data
arrayB BYTE    00h, 10h, 20h
arrayW WORD    1000h, 2000h, 3000h
arrayDW DWORD  1111h, 2222h, 3333h
ptrB  DWORD arrayB
ptrW  DWORD arrayW
ptrDW DWORD arrayDW

.code
mov al, [ptrB]          ; (A) al=?    [1pt]
```

```
mov bx, [ptrW]          ; (B) bx=?    [2pt]
mov ecx, [ptrDW]        ; (C) ecx=?   [2pt]
```

- a) [5pt] What are the values of registers in (A-C) in the above code? If there is an error, write ERROR and justify your answer. Please write in hexadecimal form.
- b) [8pt] Let's suppose 'dwList LABEL DWORD' be added in the following code. What are the values of registers in (A-D) in the code below? If there is an error, write ERROR and justify your answer. Please write in hexadecimal form.

```
.data
dwList LABEL DWORD
arrayW WORD 1000h, 2000h, 3000h
arrayDW DWORD 1111h, 2222h, 3333h
...
.code
mov eax, dwList          ; (A) eax=?    [2pt]
mov ebx, [dwList + 1]    ; (B) ebx=?    [2pt]
mov ecx, [dwList + 2]    ; (C) ecx=?    [2pt]
mov edx, [dwList + 3]    ; (D) edx=?    [2pt]
```

- 3) [4pt] What will be the value of EAX and the Sign flag, Overflow flag, Carry flag, after the following lines execute?

```
mov eax, 1
sub eax, 129              ; eax= , SF= , OF= , CF=
```

- 4) [4pt] What will be the value of AL and the Sign flag, Overflow flag, Carry flag, after the following lines execute?

```
mov al, 5
add al, 123               ; al= , SF= , OF= , CF=
```

3. Understanding Assembly Programming (14 points)

```
.data
ArrayD1      DWORD      "1024", '9aB0', -128
ArrayB       BYTE       32h, 31, 1 DUP(-15)
ArrayW1      WORD       '31', 23, 0AAFFh, -20, -30
ArrayW2      WORD       10 DUP(-12), 40, "40", -40
ArraySize = $-ArrayW1
ArrayD2      DWORD      '9ba0', ArraySize, -256
```

Suppose the following program executes with respect to data declaration above. Write the output of destination operand for every instruction execution. If an instruction is wrong, then, mark 'X'. Also, assume that the preceding instruction does not affect the following instruction. Use supplementary material for ASCII table if necessary. Also, assume Little Endian. (Each line is 2 points.)

movsx	ax, [ArrayB+2]	; ax=?
movzx	ax, [ArrayB+8]	; ax=?
mov	ax, ArrayW1	; ax=?
mov	eax, WORD PTR ArrayD2	; eax=?
mov	eax, [ArrayD2+4]	; eax=?
mov	eax, [ArrayD2-4]	; eax=?
mov	ax, [ArrayD1+0Fh]	; ax=?

4. Write in Assembly Language-1 (15 points)

The following code is to reverse the elements of the 'WORD type array' named arr. Please fill in the blanks below. (Hint: Use the xchg instruction, SIZEOF, LENGTHOF, TYPE operators to fill in the blanks.) (Each line is 3 points.)

```

; input array name: arr (WORD)
mov eax, 0 ; eax is used for index.
mov ebx, SIZEOF arr - TYPE arr ; ebx is also used for index,
mov ecx, ____

L1:
    mov dx, arr[eax] ; dx is temporary storage.
    ____ dx, ____
    mov ____, dx
    add eax, ____
    sub ebx, ____
    loop L1

```

5. Write in Assembly Language-2 (10 points)

```

.data
FinalResult DWORD 11223344h

.code
mov al, 4

```

```

mov bl, 3
mov esi, OFFSET FinalResult
mov ecx, 4

L1:
    mov BYTE PTR [esi], al
    sub BYTE PTR [esi], bl
    mov al, bl
    mov bl, BYTE PTR [esi]
    inc esi
    loop L1

```

Answer each question:

- (a) [5pt] What is the FinalResult value after the final execution in the above code? Please write in hexadecimal form in little endian.
- (b) [5pt] The third line of the above code segment 'mov esi, OFFSET FinalResult' has been changed to 'mov esi, 0'. Modify the code in the loop to get the same result.

6. Write in Assembly Language-3 (12 points)

The following code interchanges the numbers in odd-numbered places and even-numbered places in arrayB1 and store the interchanged values in arrayB2. (i.e., interchange values at 1st place and 2nd place). Please fill in the blanks below. (Hint: Use LEGNTHOF to fill in the blanks.) (Each line is 3 points.)

```

.data
arrayB1 BYTE 00h, 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 09h
arrayB2 BYTE 10 DUP(?)

.code
mov edx, 0; edx is used for index.
mov ecx, ___ ; loop count
L1:
    mov __, __ ; store arrayB1[i] into [ ] register
    mov __, __ ; store arrayB1[i + 1] into [ ]
    mov __, ax ; use PTR operator
    add edx, 2
    loop L1

```