In-Class, Open Book Mid-Term Examination

March 23, 2021

The work on this examination is to be your own and you are expected to adhere to the UMass-Boston honor system. All questions can be answered by one or two short sentences. Do not try to make up for a lack of understanding by providing a rambling answer.

**Note: I give partial credit! Show all work!**

1. (20 points) Short Questions

1. (2 points) What is the use of the Control Bus?

The control bus is used to communicate between the cpu and other internal devices

1. (2 points) If %eax = 0x100200, what is wrong with the following Intel assembly instruction?

cmpl 8(%eax), 0x100100

improper usage, cant compare to an address

1. (4 points) Name an advantage and a disadvantage of a processor that has more bits in its address bus and data bus over another processor that has fewer bits on these busses?

Advantage: more addressable locations / more info can be transferred at once

Disadvantage: complexity cost

1. (2 points) What is the difference between dynamic RAM and static RAM

DRAM only retains memory briefly and needs refreshing, SRAM maintains memory until power loss

1. (6 points) For the following program:

void main( ){ char \* ptr; \*ptr = 5;}

i) Which part of the memory can you find ptr?

The stack

ii) What does the program do when you run it on the linux server?

It segfaults, the memory address is used by the code and linux will not allow you make changes there

iii) What does the program do when you run it on the tutor VM?

It will allow you to change this value, as the tutor doesn’t care where you access and write when running code

1. (4 points) In Lab 2, why do you use the function pgm\_read\_byte(&a[i]) to read the EEPROM content instead of just using a[i]?

This is due to the eeprom not being accessibly the same way

2. (20 points) Evaluations

a. (10 points) The content of 16 memory locations starting at 0x00100250 is:

00100250 01 23 45 67 58 02 10 00

89 ab cd ef 00 01 10 11

Show the content of %ecx, %eax after

executing these instructions:

movl 0x00100254, %ecx

movl 4(%ecx), %eax

%ecx= \_\_\_0x00100254\_\_\_\_\_\_\_\_\_\_\_\_

%eax = \_\_\_fe cd ab 89\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b. (10 points) Show the hex value of the eax register and state of the specified condition flags after executing the instruction.

movl $0xf0f0f0f0, %eax

movl $0x0f0f0f0f, %ecx

xorl %ecx %eax

%eax = 0x11111111

CF = \_\_0\_\_

SF = \_1\_\_\_

ZF = \_\_0\_\_

OF = \_0\_\_\_

3. (20 points) Stack Operations

You are debugging a program on the tutor VM. The content of 16 memory locations starting at 0x003fffe0 is:

003fffe0 bc f0 10 00 00 bd 89 ab

00 00 e8 01 00 00 00 cc

%eax = 0x12345678

%esp = 0x003fffe4

What are the memory and register content after you execute the instruction: pushl %eax ?

0x003fffe0: \_\_bc\_\_\_\_\_\_

0x003fffe1: \_\_\_f0\_\_\_\_\_

0x003fffe2: \_\_10\_\_\_\_\_\_

0x003fffe3: \_\_\_00\_\_\_\_\_

0x003fffe4: \_\_\_78\_\_\_\_\_

0x003fffe5: \_\_\_56\_\_\_\_\_

0x003fffe6: \_\_\_43\_\_\_\_\_

0x003fffe7: \_\_\_12\_\_\_\_\_

%eax = \_\_\_0x12345678\_\_\_\_\_\_\_\_

%esp = \_\_\_0x003fffe8\_\_\_\_\_\_\_\_\_

4. (40 points) Assembly language program

Write a C callable assembly language function

(change\_case.s) to change the alphabets in a string to either upper case or lower case. The user will select an option. The change\_case function will return the string with the correct case or it will return the error string if an invalid option is selected.

The function prototype of the assembly language function in C is shown as:

extern char \* change\_case(char option);

Your assembly language function should get the option from a C main function shown below:

/\* change\_casec.c: C driver for the changing case

function. Users can enter an option:

option = ‘U’ or 0x55: change all to upper case

The function will return the error string if an

invalid option is selected.

\*/

#include <stdio.h>

extern chat \* change\_case(char option);

int main()

{

char option;

char \*ptr;

printf("Enter case change option : ");

scanf(“%c”, &option);

ptr = change\_case(option);

printf("\nThe new string is : %s\n", ptr);

return 0;

}

# Change case assembly language program

**# You only have to implement the option = ‘U’**

**# Return the error string if option != ‘U’**

**# You can find the ascii code chart at the end**

#

.data

list: .asciz “**All Good Things Come To Those Who Wait”**

error: .asciz “Invalid option”

.text

.globl change\_case

change\_case:

push %ebp

mvl %esp, ebp

subl $4, %esp

mvl 8(ebp), %eax

movl list, %ecx

compb %eax, 0x55

jne do-error:

loop:

compb $0, (%ecx)

je done

movl $0, ebx ,

subl $20, (%ecx)

movb ebx(%ecx), %edx

addb $1, %edx

addl $1, %ebx

do-error:

movl error, %edx

done:

movl, %ebp, %esp

popl, %esp

ret .end

.end

**ASCII Code Chart:**

Here is the **ASCII Encoding**, a correspondence of keyboard characters with integers from 0 to 127 (0x7F in hexadecimal, 0177 in octal)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| char | hex | oct |  | char | hex | oct |  | char | hex | oct |  | char | hex | oct |
| NUL | 00 | 000 |  | SP | 20 | 040 |  | @ | 40 | 100 |  | ` | 60 | 140 |
| SOH | 01 | 001 |  | ! | 21 | 041 |  | A | 41 | 101 |  | a | 61 | 141 |
| STX | 02 | 002 |  | " | 22 | 042 |  | B | 42 | 102 |  | b | 62 | 142 |
| ETX | 03 | 003 |  | # | 23 | 043 |  | C | 43 | 103 |  | c | 63 | 143 |
| EOT | 04 | 004 |  | $ | 24 | 044 |  | D | 44 | 104 |  | d | 64 | 144 |
| ENQ | 05 | 005 |  | % | 25 | 045 |  | E | 45 | 105 |  | e | 65 | 145 |
| ACK | 06 | 006 |  | & | 26 | 046 |  | F | 46 | 106 |  | f | 66 | 146 |
| BEL | 07 | 007 |  | ' | 27 | 047 |  | G | 47 | 107 |  | g | 67 | 147 |
| BS | 08 | 010 |  | ( | 28 | 050 |  | H | 48 | 110 |  | h | 68 | 150 |
| HT | 09 | 011 |  | ) | 29 | 051 |  | I | 49 | 111 |  | i | 69 | 151 |
| NL/LF | 0A | 012 |  | \* | 2A | 052 |  | J | 4A | 112 |  | j | 6A | 152 |
| VT | 0B | 013 |  | + | 2B | 053 |  | K | 4B | 113 |  | k | 6B | 153 |
| NP/FF | 0C | 014 |  | , | 2C | 054 |  | L | 4C | 114 |  | l | 6C | 154 |
| CR | 0D | 015 |  | - | 2D | 055 |  | M | 4D | 115 |  | m | 6D | 155 |
| SO | 0E | 016 |  | . | 2E | 056 |  | N | 4E | 116 |  | n | 6E | 156 |
| SI | 0F | 017 |  | / | 2F | 057 |  | O | 4F | 117 |  | o | 6F | 157 |
| DLE | 10 | 020 |  | 0 | 30 | 060 |  | P | 50 | 120 |  | p | 70 | 160 |
| DC1 | 11 | 021 |  | 1 | 31 | 061 |  | Q | 51 | 121 |  | q | 71 | 161 |
| DC2 | 12 | 022 |  | 2 | 32 | 062 |  | R | 52 | 122 |  | r | 72 | 162 |
| DC3 | 13 | 023 |  | 3 | 33 | 063 |  | S | 53 | 123 |  | s | 73 | 163 |
| DC4 | 14 | 024 |  | 4 | 34 | 064 |  | T | 54 | 124 |  | t | 74 | 164 |
| NAK | 15 | 025 |  | 5 | 35 | 065 |  | U | 55 | 125 |  | u | 75 | 165 |
| SYN | 16 | 026 |  | 6 | 36 | 066 |  | V | 56 | 126 |  | v | 76 | 166 |
| ETB | 17 | 027 |  | 7 | 37 | 067 |  | W | 57 | 127 |  | w | 77 | 167 |
| CAN | 18 | 030 |  | 8 | 38 | 070 |  | X | 58 | 130 |  | x | 78 | 170 |
| EM | 19 | 031 |  | 9 | 39 | 071 |  | Y | 59 | 131 |  | y | 79 | 171 |
| SUB | 1A | 032 |  | : | 3A | 072 |  | Z | 5A | 132 |  | z | 7A | 172 |
| ESC | 1B | 033 |  | ; | 3B | 073 |  | [ | 5B | 133 |  | { | 7B | 173 |
| FS | 1C | 034 |  | < | 3C | 074 |  | \ | 5C | 134 |  | | | 7C | 174 |
| GS | 1D | 035 |  | = | 3D | 075 |  | ] | 5D | 135 |  | } | 7D | 175 |
| RS | 1E | 036 |  | > | 3E | 076 |  | ^ | 5E | 136 |  | ~ | 7E | 176 |
| VS | 1F | 037 |  | ? | 3F | 077 |  | \_ | 5F | 137 |  | DEL | 7F | 177 |