

CPE 323 Introduction to Embedded Computer Systems

Homework IV

1 (25)	2 (20)	Total
—	—	—

Problem #1. (25 points) C language, Stack Data Placement, Pointers

Consider the following C program. Assume that the register SP at the beginning points to 0x0E00. Answer the following questions. Assume all variables are allocated on the stack, and in the order as they appear in the program.

A. (9 points) Illustrate the content of the stack at the moment before the statement at line 7 is executed.

B. (9 points) Comment the code (lines 7 – 11) indicating the result of each statement. Illustrate the content of the stack at the end of execution of the statement in line 11.

What are the contents of arrays x and carr?

C. (6 points) Show assembly language implementation of the statement at lines 5, 7 (think how would compiler translate these statements; it knows where the variables are placed relatively to the current top of the stack).

1	volatile unsigned int x[3]={1, 32768, 65533};
2	volatile char carr[4]={'C', '1', '0', 'a'};
3	volatile long int z = 65540;
4	volatile char *p_c = carr; //
5	volatile unsigned int *p_i = x; //
6	
7	*(p_c + 2) += '6'; //
8	p_i = p_i + 3; //
9	*p_i += *(p_i -6); //
10	p_c = p_c + 3; //
11	*p_c = 'A'; //

[illegible]

B.

1	volatile unsigned int x[3]={1, 32768, 65533};
2	volatile char carr[4]={'C', '1', '0', 'a'};
3	volatile long int z = 65540;
4	volatile char *p_c = carr; //
5	volatile unsigned int *p_i = x; //
6	
7	*(p_c + 2) += '6'; //
8	p_i = p_i + 3; //
9	*p_i += *(p_i -6); //
10	p_c = p_c + 3; //
11	*p_c = 'A'; //

Address	M[15..0]	Comment

C.

; volatile unsigned int *p_i = x

*(p_c + 2) += '6';

Problem #2. (20 points) C language, Stack Data Placement, Pointers

Consider the following C program. Assume that the register SP at the beginning points to 0x0A00. Answer the following questions. Assume all variables are allocated on the stack, and in the order as they appear in the program. ASCII code for character '0' is 48.

1	int main(void) {
2	volatile long int a = -67;
3	volatile int c = -6, d = -5;
4	volatile char mych[4] = { '0', '2', '4', '8' };
5	volatile long int *lpa = &a;
6	volatile int *pi = &d;
7	lpa = lpa - 2; //
8	*lpa = *lpa + 1026; //
9	pi += 2; //
9	*pi = *pi + c; //
11	}

Fill in the following table by determining the values/addresses given below.

#	Question?	Value/Address
1	The number of bytes allocated on the stack for the variable declared in line 2.	
2	The number of bytes allocated on the stack for the character array declared in line 4.	
3	The number of bytes allocated on the stack for all variables declared in lines 2-6.	
4	Value of mych[0] after initialization performed in line 4.	
5	Address of variable a (&a).	
6	Value of lpa at the moment after the statement in line 7 is executed.	
7	Value of *lpa at the moment after the statement in line 8 is executed.	
8	Value of mych[0] at the moment after the statement in line 8 is executed.	
9	Value of pi at the moment after the statement in line 9 is executed.	
10	Value of *pi at the moment after the statement in line 10 is executed.	

(Note: The table below is not going to be graded; you can use it to sketch the stack if you want)

[illegible]